



TUGAS AKHIR - TE 145561

**RANCANG BANGUN ALAT PEMUTAR NARASI AUDIO
TENTANG OBYEK PADA MUSEUM DENGAN
MENGUNAKAN TEKNOLOGI RFID**

Cahyo Dhian Tyastono
NRP 2214030025
Dany Prasetyo
NRP 2214030108

Dosen Pembimbing
Dr.Ir. Hendra Kusuma, M.Eng.Sc

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***AUDIO NARRATION PLAYER DEVICE DESIGN FOR
MUSEUM OBJECT USING RFID TECHNOLOGY***

Cahyo Dhian Tyastono
NRP 2214030025
Dany Prasetyo
NRP 2214030108

Supervisor
Dr.Ir. Hendra Kusuma, M.Eng.Sc

***COMPUTER CONTROL STUDY PROGRAM
DEPARTEMENT OF ELECTRICAL ENGINEERING AUTOMATION
Vocational Faculty
Sepuluh Nopember Insitute of Technology
Surabaya 2017***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Rancang Bangun Alat Pemutar Narasi Audio Tentang Obyek Pada Museum Dengan Menggunakan Teknologi RFID”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.


Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Cahya Dhin Tyastono
NRP 2214030025

Surabaya, 18 Juli 2017



Dany Prasetyo
NRP 2214030108

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN ALAT PEMUTAR NARASI AUDIO
TENTANG OBYEK PADA MUSEUM DENGAN
MENGUNAKAN TEKNOLOGI RFID**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Ahli Madya Teknik Elektro**

**Pada
Bidang Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui

Dosen Pembimbing

Dr. Ir. Hendra Kusuma, M. Eng. Sc

NIP. 19640902 198903 1 003

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN ALAT PEMUTAR NARASI AUDIO TENTANG OBYEK PADA MUSEUM DENGAN MENGUNAKAN TEKNOLOGI RFID

Nama : Cahyo Dhian Tyastono
NRP : 2214030025
Nama : Dany Prasetyo
NRP : 2214030108
Pembimbing : Dr.Ir. Hendra Kusuma, M.Eng.Sc
NIP : 196409021989031003

ABSTRAK

Secara umum informasi yang diperoleh oleh pengunjung pada museum diberikan secara tertulis di sekitar obyek maupun diterangkan oleh pemandu museum dan belum dilakukan secara otomatis dengan menggunakan perangkat elektronik sehingga kurang efektif dalam memperoleh informasi dari obyek. Maka pada Tugas Akhir ini dilakukan perancangan suatu perangkat yang menggunakan teknologi RFID (*Radio Frequency Identification Device*) yang dapat memainkan suara narasi untuk memberi penjelasan mengenai informasi dari obyek pada museum sesuai dengan kebutuhan pengunjung.

Perancangan alat pemutar suara narasi pada obyek museum ini terdiri dari modul *reader* MFRC522, *Arduino Mega 2560*, LCD 16×4 dan *MP3 Shield*. *Reader* MFRC522 digunakan untuk membaca *UID tag* dan *Arduino* sebagai pengendali utama dalam sistem yang dapat mengenali obyek berdasarkan pembacaan *UID tag* dan mengendalikan modul *MP3 Shield* agar dapat memainkan suara penjelasan yang sesuai dengan obyek museum. Aplikasi basis data menggunakan basis data *visual basic* untuk melihat daftar obyek yang telah disimak pengunjung.

Hasil pengambilan data pada modul *reader* terhadap *tag* hanya mampu membaca *UID tag* dengan jarak maksimal 3cm. Perangkat yang dibuat mampu memberikan informasi tentang obyek museum dalam bentuk suara narasi dengan 2 pilihan bahasa yaitu Bahasa Indonesia dan Bahasa Inggris. Aplikasi basis data yang digunakan mampu menerima data dari perangkat keras dan menampilkannya pada tabel, data tersebut berupa tanggal, jam, menit dan nama obyek yang telah simak pengunjung.

Kata Kunci : RFID, Obyek museum, *Arduino Mega 2560*, *Visual basic*

-----Halaman ini sengaja dikosongkan-----

AUDIO NARRATION PLAYER DEVICE DESIGN FOR MUSEUM OBJECT USING RFID TECHNOLOGY

Name : Cahyo Dhian Tyastono
Register Number : 2214030025
Name : Dany Prasetyo
Register Number : 2214030108
Supervisor : Dr.Ir. Hendra Kusuma, M.Eng.Sc
ID Number : 196409021989031003

ABSTRACT

In general the information obtained by visitors to the museum is given in writing around the object or explained by the museum guide and has not been done automatically by using electronic devices so less effective in obtaining information from the object. So in this Final Project is a device that uses RFID technology (Radio Frequency Identification Device) is designed which can play the narration voice to give explanation about the information of objects in the museum according to the needs of visitors.

The design of the narration voice player on the museum object consists of MFRC522 reader module, Arduino Mega 2560, 16 × 4 LCD and MP3 Shield. The MFRC522 Reader is used to read tag's UID and Arduino as the main controller in the system that can recognize the objects based on tag's UID reading and control the MP3 Shield module in order to play explanation voice which appropriate to the museum objects. A visual basic database application is used to see the list of the objects that have been viewed by the visitors.

The data retrieval results in the reader module to the tags are only able to read tag's UID with a maximum distance of 3cm. The created device is able to provide information about the museum object in the form of narrative sound with 2 choices of languages that is Indonesian and English. Database application which used is able to receive data from the hardware and display it in the table, the data in the form of date, hour, minute and the name of the object that has been seen by visitors.

Keywords : RFID, Museum Object, Arduino Mega 2560, Visual Basic

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Jurusan D3 Teknik Elektro, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN ALAT PEMUTAR NARASI AUDIO TENTANG OBYEK PADA MUSEUM DENGAN MENGUNAKAN TEKNOLOGI RFID

Kami mengucapkan terimakasih kepada semua pihak yang telah mendukung dalam pembuatan buku tugas akhir ini, termasuk dosen pembimbing kami, yaitu Dr.Ir. Hendra Kusuma, M.Eng.Sc yang telah menyempatkan waktunya untuk membimbing kami dalam menyelesaikan tugas akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 18 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	ii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Sistematika Laporan.....	4
1.6 Relevansi.....	5
BAB II TEORI DASAR	7
2.1 <i>Radio Frequency Identification Device (RFID)</i>	7
2.1.1 <i>Tag RFID</i>	8
2.1.2 <i>Reader RFID</i>	10
2.2 Sistem dan Cara Kerja RFID	10
2.3 <i>Push Button</i>	12
2.4 <i>Buzzer</i>	14
2.5 LCD 16x4	14
2.6 Modul 12C LCD	16
2.7 RTC (<i>Real Time Clock</i>)	17
2.7.1 RTC DS1307	17
2.8 Mifare RC522 RFID Reader Module	19
2.9 MP3 Shield	20
2.10 Arduino Mega 2560	22
2.10.1 Pemrograman Arduino	27
2.11 Visual Basic 6.0	28

2.12	Obyek Museum	30
2.12.1	<i>Map of Hell</i>	31
2.12.2	<i>The Scream</i>	31
2.12.3	<i>La Scapigliata</i>	32
2.12.4	Patung <i>Liberty</i>	33
2.12.5	Menara Eiffel.....	34
BAB III PERANCANGAN DAN PEMBUATAN ALAT		37
3.1	Gambaran Umum Sistem.....	37
3.2	Perancangan Perangkat Keras.....	39
3.2.1	Pengkabelan <i>Arduino Mega 2560</i> dengan <i>Reader MFRC522</i>	39
3.2.2	Rangkaian RTC	41
3.2.3	Pengkabelan Rangkaian RTC dengan <i>Arduino</i> <i>Mega 2560</i>	41
3.2.4	Pengkabelan LCD dan I2C dengan <i>Arduino Mega</i> <i>2560</i>	42
3.2.5	Rangkaian <i>Keypad</i>	43
3.2.6	Pengkabelan <i>Keypad</i> dengan <i>Arduino Mega 2560</i>	44
3.2.7	Pengkabelan <i>MP3 Shield</i> dengan <i>Arduino Mega 2560</i>	46
3.2.8	Perancangan <i>Design Box</i>	47
3.2.9	Perancangan <i>Display</i> Obyek Museum.....	51
3.3	Perancangan Perangkat Lunak.....	52
3.3.1	<i>Flowchart</i> Sistem <i>Scanning Tag</i> RFID.....	53
3.3.2	<i>Flowchart</i> Perancangan Sistem Pemilihan Bahasa dan Menyimpan Nama Obyek ke <i>EEPROM Arduino</i> ..	54
3.3.3	<i>Flowchart</i> Perancangan Sistem untuk Menghentikan <i>MP3 Shield</i>	54
3.3.4	<i>Flowchart</i> Prosedur <i>Set-Up</i> Komunikasi Serial <i>Arduino</i> dengan <i>PC</i>	55
3.3.5	<i>Flowchart</i> untuk Menampilkan Data Nama Obyek pada Antarmuka <i>Visual Basic</i>	55
3.3.6	Pembuatan Antarmuka Menggunakan <i>Visual Basic</i> ..	60
BAB IV PENGUKURAN DAN PENGUJIAN		63
4.1	Pengukuran dan Pengujian RFID	63
4.2	Pengujian RFID dengan LCD 16x4 untuk Menampilkan Nama Obyek	65
4.3	Pengujian <i>Arduino Mega 2560</i> dengan <i>MP3 Shield</i>	67
4.4	Pengujian RTC	71

4.5	Pengujian Komunikasi Serial <i>Arduino</i> dengan PC	72
4.6	Pengujian Antarmuka <i>Visual Basic</i>	74
4.7	Pengujian Keseluruhan Alat.....	77
BAB V PENUTUP.....		81
DAFTAR PUSTAKA		83
LAMPIRAN.....		85
A.	Dokumentasi Alat	85
B.	<i>Listing</i> Program.....	89
C.	<i>Datasheet / Specification</i>	103
D.	Daftar Riwayat Hidup	121

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1	Komponen <i>Tag</i> RFID8
Gambar 2.2	Sistem Kerja RFID 11
Gambar 2.3	<i>Push Button</i> 13
Gambar 2.4	Prinsip Kerja <i>Push Button</i> 13
Gambar 2.5	Bentuk Fisik <i>Buzzer</i> 14
Gambar 2.6	Tampilan LCD 16x4..... 15
Gambar 2.7	Modul I2C LCD 16
Gambar 2.8	Konfigurasi Pin DS1307 17
Gambar 2.9	Tampilan Modul <i>Reader</i> MRC522.....20
Gambar 2.10	Bentuk Fisik Modul MP3 <i>Shield</i> 21
Gambar 2.11	Bentuk Fisik <i>Arduino Mega</i> 2560 22
Gambar 2.12	Contoh Pemrograman <i>Arduino</i> 28
Gambar 2.13	Tampilan Lingkungan Kerja <i>Visual Basic</i> 6.0.....29
Gambar 2.14	Lukisan <i>Map of Hell</i>31
Gambar 2.15	Lukisan <i>The Scream</i>32
Gambar 2.16	Lukisan <i>La Scapigliata</i>33
Gambar 2.17	Patung <i>Liberty</i>34
Gambar 2.18	Menara Eiffel.....35
Gambar 3.1	Arsitektur Sistem.....37
Gambar 3.2	Diagram Fungsional Keseluruhan Sistem38
Gambar 3.3	Skema Pengkabelan <i>Arduino Mega</i> 2560 dengan <i>Reader</i> MFRC522.....40
Gambar 3.4	Rangkaian RTC DS130741
Gambar 3.5	Skema Pengkabelan RTC dengan <i>Arduino Mega</i> 2560...42
Gambar 3.6	Pengkabelan LCD, I2C dengan <i>Arduino Mega</i> 256043
Gambar 3.7	Skema Rangkaian <i>Keypad</i>44
Gambar 3.8	Pengkabelan <i>Keypad</i> dengan <i>Arduino Mega</i> 2560.....45
Gambar 3.9	Skematik Modul MP3 <i>Shield</i> dengan <i>Arduino Mega</i> 2560.....47
Gambar 3.10	Perancangan <i>Box</i> Tampak Depan48
Gambar 3.11	Perancangan <i>Box</i> Tampak Belakang49
Gambar 3.12	Perancangan <i>Box</i> Tampak Atas dan Tampak Bawah49
Gambar 3.13	Perancangan <i>Box</i> Tampak Samping50

Gambar 3.14	Perancangan <i>Box</i> Keseluruhan Ketika Disatukan.....	50
Gambar 3.15	Perancangan <i>Display</i> Obyek Tampak Depan	51
Gambar 3.16	Perancangan <i>Display</i> Obyek Tampak Belakang.....	52
Gambar 3.17	Perancangan <i>Display</i> Obyek Tampak Samping.....	52
Gambar 3.18	<i>Flowchart</i> Sistem <i>Scanning Tag</i> RFID.....	56
Gambar 3.19	<i>Flowchart</i> Sistem Pemilihan Bahasa	57
Gambar 3.20	<i>Flowchart</i> Sistem Menghentikan MP3 <i>Shield</i>	58
Gambar 3.21	<i>Flowchart</i> Sistem Komunikasi Serial Antara <i>Arduino</i> dengan PC dengan <i>platform</i> VB	59
Gambar 3.22	<i>Flowchart</i> Menampilkan Data Obyek Pada Antarmuka VB	60
Gambar 3.23	Perancangan Antarmuka pada <i>Form Designer</i>	61
Gambar 4.1	Skema Pengambilan Data jarak Baca Modul <i>Reader</i> pada <i>Tag</i>	63
Gambar 4.2	Pengujian Pengambilan UID <i>Tag</i> dalam jarak 1-3 cm ...	64
Gambar 4.3	Skema untuk Pengujian RFID dengan LCD 16x4 untuk Menampilkan Nama Obyek.....	65
Gambar 4.4	<i>Listing</i> Program untuk Pengujian <i>Arduino Mega 2560</i> dengan MP3 <i>Shield</i>	68
Gambar 4.5	Replika Obyek Museum Untuk Pengambilan Data	69
Gambar 4.6	Program <i>Void</i> jam.....	71
Gambar 4.7	<i>Listing</i> Program untuk Pengiriman Data dari EEPROM <i>Arduino</i> untuk Komunikasi Serial	73
Gambar 4.8	Tampilan Awal Aplikasi Antarmuka Data Obyek Museum	75
Gambar 4.9	Tampilan Aplikasi Antarmuka Setelah Komunikasi Disambungkan.....	76
Gambar 4.10	Pengujian Alat Pemutar Narasi.....	77

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Karakteristik <i>Tag</i> RFID.....	10
Tabel 2.2 Fungsi Pin I/O DS1307	18
Tabel 2.3 Spesifikasi MRC522	20
Tabel 2.4 Karakteristik Modul MP3 <i>Shield</i>	21
Tabel 2.5 Spesifikasi <i>Arduino Mega 2560</i>	22
Tabel 4.1 Hasil Data Jarak Pembacaan <i>Tag</i>	64
Tabel 4.2 Hasil Pembacaan UID <i>Tag</i> Pada Jarak 1cm.....	66
Tabel 4.3 Hasil Pembacaan UID <i>Tag</i> Pada Jarak 2cm.....	66
Tabel 4.4 Hasil Pembacaan UID <i>Tag</i> Pada Jarak 3cm.....	67
Tabel 4.5 Data Obyek Pada SD-card MP3 <i>Shield</i>	69
Tabel 4.6 Hasil Pengujian Antara RFID dengan Obyek	70
Tabel 4.7 Hasil Pengujian Waktu RTC dengan Waktu pada Laptop	72
Tabel 4.8 Pengujian Komunikasi Serial <i>Arduino</i> dengan PC.....	73
Tabel 4.9 Hasil Pengujian Keseluruhan Sistem Alat Pemutar Narasi Obyek.....	78

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang pembuatan Tugas Akhir, perumusan masalah, Batasan masalah, tujuan, sistematika penulisan, dan relevansi.

1.1 Latar Belakang

Untuk mengetahui tentang suatu sejarah, menambah ilmu pengetahuan, ataupun sekedar berjalan-jalan, banyak orang yang akan berkunjung ke museum. Terdapat berbagai macam museum dengan berbagai barang/obyek koleksinya di seluruh dunia ini yang bermacam-macam. Barang-barang koleksi yang berada di museum tidak hanya sekedar untuk dijadikan pajangan dan dilihat-lihat saja oleh pengunjung, melainkan barang-barang koleksi tersebut akan diamati dan juga dipelajari oleh pengunjung yang memang berniat mengetahui tentang sejarah atau cerita di balik koleksi yang terdapat di museum ketika sedang berkunjung di museum. Oleh karena itu, museum memberikan narasi atau penjelasan yang tertulis di dekat barang-barang koleksinya agar pengunjung mengetahui sejarah atau deskripsi dari koleksi yang dipamerkan. Namun, narasi yang berupa teks tertulis tersebut masih memiliki beberapa kekurangan. Diantaranya, narasi berupa teks tulisan tersebut mungkin akan kurang efisien karena dengan narasi tertulis tersebut fokus pengunjung pada barang koleksi akan sedikit terganggu karena pengunjung harus membaca teks untuk mengetahui deskripsi dari barang koleksi tersebut. Masalah lainnya adalah apabila narasi yang sedikit akan mengurangi informasi dari barang koleksi, tetapi jika narasi tertulis secara panjang lebar, pengunjung mungkin akan merasa bosan dengan panjangnya narasi yang harus dibaca.

Untuk itu banyak museum-museum yang menyediakan pemandu museum untuk memandu pengunjung ketika berkeliling menikmati koleksi yang ada di museum. Pemandu museum akan membantu menerangkan informasi dari barang-barang koleksi pada museum. Ulasan dari pemandu akan lebih menarik untuk didengarkan oleh pengunjung daripada hanya membaca narasi tertulis pada koleksi museum. Informasi yang diutarakan oleh pemandu pada pengunjung biasanya juga akan lebih jelas. Namun hal tersebut juga tidak lepas dari masalah. Pemandu museum memang efektif untuk menceritakan

informasi koleksi museum ketika pengunjungnya tidak terlalu banyak. Pasalnya, pengunjung pada museum seringkali tidak datang dalam jumlah sedikit. Biasanya untuk suatu *tour* museum, pengunjung akan datang secara berkelompok dalam jumlah yang banyak dan sumber daya pemandu pada museum biasanya hanya terdiri dari beberapa orang saja. Jika ada banyak kelompok *tour* yang datang untuk mengunjungi museum, pemandu akan kesulitan untuk memandu dalam waktu yang bersamaan dan hasilnya akan terasa kurang efektif. Narasi yang disampaikan oleh pemandu akan kurang jelas dengan banyaknya pengunjung yang mendengarkan serta pemandu akan kesulitan menangani banyaknya pertanyaan yang datang dari pengunjung. Hal itu akan berdampak pada berkurangnya informasi yang didapat dari penjelasan yang diutarakan oleh pemandu dan akan mengurangi nilai-nilai sejarah dari koleksi yang ada di museum. Pihak museum dapat menambah jumlah pemandu untuk menanggulangi hal tersebut, namun akan membutuhkan tenaga manusia yang banyak dan akan menambah pengeluaran dari pihak museum untuk biaya yang bertambah dari jasa pemandu museum.

Oleh karena itu pada tugas akhir ini kami merancang suatu alat yang memanfaatkan RFID (*Radio Frequency Identification*) dan mikrokontroler serta dilengkapi dengan Mp3 *shield* untuk alat yang berfungsi sebagai pemandu museum *virtual*. Sistem menggunakan RFID untuk mendeteksi barang-barang koleksi yang ada di museum sehingga sistem ini juga bisa digunakan sebagai sistem pengaman barang-barang koleksi dari museum tersebut. Setiap koleksi pada museum memiliki *tag* RFID masing-masing untuk menandai masing-masing barang koleksi sesuai dengan narasinya yang nantinya akan dideteksi dengan RFID *Reader*. Sistem ini memanfaatkan mikrokontroler sebagai otak. Mp3 *shield* memiliki fungsi untuk memutar narasi dari setiap koleksi museum yang telah dideteksi dengan RFID.

1.2 Permasalahan

Melihat latar belakang di atas, maka dirumuskan permasalahan yaitu penggunaan tulisan pada obyek-obyek di museum kurang efisien karena dengan narasi tertulis tersebut fokus pengunjung pada barang koleksi akan sedikit terganggu karena pengunjung harus membaca teks untuk mengetahui deskripsi dari barang koleksi tersebut dan apabila narasi yang sedikit akan mengurangi informasi dari barang koleksi, tetapi jika narasi tertulis secara panjang lebar, pengunjung mungkin

akan merasa bosan dengan panjangnya narasi yang harus dibaca. Selain itu, dengan penggunaan pemandu yang biasanya menjelaskan di museum, maka pengunjung akan mendengarkan secara bersamaan dan ramai, sehingga pengunjung akan kurang fokus dengan narasi yang dijelaskan oleh pemandu.

1.3 Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini meliputi:

1. Menggunakan lima obyek museum dengan lima *tag* RFID pada masing-masing obyek.
2. Bahasa yang digunakan dalam narasi barang-barang koleksi museum adalah bahasa Indonesia dan bahasa Inggris.
3. Mikrokontroler yang digunakan pada alat ini adalah *Arduino Mega 2560* yang mengendalikan *MP3 Shield* dan LCD berdasarkan data dari pembacaan RFID pada objek.
4. LCD menampilkan nama dari obyek museum berdasarkan dari pembacaan RFID.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini meliputi:

1. Merancang suatu alat yang memanfaatkan RFID (*Radio Frequency Identification*) dan mikrokontroler serta dilengkapi dengan *MP3 Player* untuk alat yang berfungsi sebagai pengganti pemandu museum (Penanggung jawab Dany Prasetyo).
2. Sistem menggunakan RFID untuk mendeteksi barang-barang koleksi yang ada di museum. Setiap koleksi pada museum memiliki *tag* RFID masing-masing untuk menandai masing-masing barang koleksi sesuai dengan narasinya yang nantinya akan dideteksi dengan *RFID Reader* (Penanggung jawab Dany Prasetyo).
3. Sistem ini memanfaatkan mikrokontroler sebagai otak. *MP3 Player* memiliki fungsi untuk memutar narasi dari setiap koleksi museum yang telah dideteksi dengan RFID (Penanggung jawab Cahyo Dhian Tyastono).
4. Merancang suatu sistem berbasis data dari barang-barang koleksi yang ada di museum untuk menarasikan informasi dari obyek museum dengan format MP3 (Penanggung jawab Cahyo Dhian Tyastono).

1.5 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, sistematika laporan, dan relevansi.

BAB II TEORI DASAR

Bab ini menjelaskan tentang tinjauan pustaka, RFID yang terdiri dari *reader* dan *tag*, sistem dan cara kerja RFID, MP3 *shield*, *push button*, dan buzzer yang merupakan tanggung jawab dari Dany Prasetyo serta LCD 16x4, modul I2C LCD, RTC (*Real Time Clock*), dan konsep dari *Arduino Mega 2560*, dan basis data *Visual Basic* yang merupakan tanggung jawab Cahyo Dhian Tyastono.

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Bab ini membahas tentang penjelasan dari metodologi yang digunakan pada alat pemutar narasi yang terdiri dari pembuatan perangkat keras dan pembuatan perangkat lunak yang akan digunakan untuk proses identifikasi.

BAB IV PENGUKURAN DAN PENGUJIAN

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian jarak *reader* RFID yang dapat berkeja, pengujian modul MP3 *shield* RFID dengan *Arduino* pengujian antarmuka *Visual Basic* dan pengujian keseluruhan alat. Pada tiap pengujian akan ada analisis terkait metode yang digunakan.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Proses pengerjaan Tugas Akhir ini diharapkan memberikan manfaat berupa penerapannya pada tempat-tempat yang masih menggunakan tulisan dalam penjelasan objek yang disajikan khususnya tempat seperti museum agar lebih menarik dan lebih efisien dalam penjelasan objek yang disajikan. Selain itu agar pengunjung museum dapat mendengarkan narasi tentang objek yang dilihat secara privat melalui perangkat yang dibawa oleh masing-masing pengunjung.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI DASAR

Pada bab ini akan dibahas mengenai dasar-dasar teori yang berhubungan dengan alat seperti teori penunjang yang terdapat 10 poin. Tinjauan pustaka terbagi menjadi 2 bagian pada masing-masing mahasiswa, Dany Prasetyo mencari bahan materi pada poin 2.1 sampai dengan poin 2.5 dan Cahyo Dhian Tyastono mencari bahan dan materi pada poin 2.6 sampai dengan poin 2.10 pada tiap materi yang dicari digunakan sebagai dasar materi untuk pembuatan alat pemutar narasi audio untuk obyek pada museum serta sebagai dasar materi yang dibuat masing-masing mahasiswa untuk pembuatan keseluruhan alat ini.

2.1 *Radio Frequency Identification Device (RFID)* [1]

RFID adalah proses identifikasi seseorang atau obyek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah *device* kecil yang disebut *tag* atau *transponder* (*Transmitter+Responder*). *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari *device* yang kompatibel, yaitu pembaca RFID. RFID adalah teknologi identifikasi yang fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam *device* yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

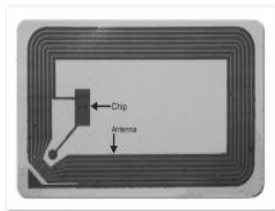
Pada sistem RFID umumnya, *tag* atau *transponder* ditempelkan pada suatu obyek. Setiap *tag* membawa dapat membawa informasi yang unik, di antaranya: serial number, model, warna, tempat perakitan, dan data lain dari obyek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada pembaca RFID, sehingga proses identifikasi obyek dapat dilakukan. Sistem RFID terdiri dari empat komponen, di antaranya *tag* adalah *device* yang menyimpan informasi untuk identifikasi obyek, *tag* RFID sering juga disebut sebagai

transponder kemudian antena untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan *tag* RFID dan Pembaca RFID adalah *device* yang kompatibel dengan *tag* RFID yang akan berkomunikasi secara *wireless* dengan *tag*. *Software* aplikasi adalah aplikasi pada sebuah *workstation* atau PC yang dapat membaca data dari *tag* melalui pembaca RFID. Baik *tag* dan pembaca RFID dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

2.1.1 Tag RFID

Tag RFID atau *transponder*, terdiri atas sebuah mikro (*microchip*) dan sebuah antena. *Chip* mikro itu sendiri dapat berukuran sekecil butiran pasir, seukuran 0,4 mm. *Chip* tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung kepada tipe memorinya. Tipe memori itu sendiri dapat *read-only*, *read-write*, atau *write-once read-many*. Antena yang terpasang pada chip mikro mengirimkan informasi dari *chip* ke *reader*. Biasanya rentang pembacaan diindikasikan dengan besarnya antena. Antena yang lebih besar mengindikasikan rentang pembacaan yang lebih jauh. *Tag* tersebut terpasang atau tertanam dalam objek yang akan diidentifikasi.

Tag RFID sangat bervariasi dalam hal bentuk dan ukuran. Sebagian *tag* mudah ditandai, misalnya *tag* anti-pencurian yang terbuat dari plastik keras yang dipasang pada barang-barang di toko. *Tag* untuk *tracking* hewan yang ditanam di bawah kulit berukuran tidak lebih besar dari bagian lancip dari ujung pensil. Bahkan ada *tag* yang lebih kecil lagi yang telah dikembangkan untuk ditanam di dalam serat kertas uang. *Tag* RFID terbagi menjadi 3 jenis, yaitu *tag* pasif, *tag* aktif, dan *tag* semipasif. Pengelempokan ini berdasarkan pada ada tidaknya catu daya pada *tag* dan kemampuannya untuk menginisiasi komunikasi dengan *reader*. Gambar 2.1 merupakan isi dari *tag* yang terdiri dari mikro dan antena dan pada Tabel 2.1 merupakan karakteristik dari *tag* RFID.



Gambar 2.1 Komponen *Tag* RFID

a. *Tag* Pasif

Tag versi paling sederhana adalah *tag* pasif, yaitu *tag* yang tidak memiliki catu daya sendiri serta tidak dapat menginisiasi komunikasi dengan *reader*. Sebagai gantinya, *tag* merespon emisi frekuensi radio dan menurunkan dayanya dari gelombang energi yang dipancarkan oleh *reader*. Sebuah *tag* pasif minimum mengandung sebuah *identifier* unik dari sebuah item yang dipasang *tag* tersebut. Data tambahan tergantung kepada kapasitas penyimpanannya. Dalam keadaan yang sempurna, sebuah *tag* dapat dibaca dari jarak sekitar 10 hingga 20 kaki. *Tag* pasif dapat beroperasi pada frekuensi rendah (*low frequency*, LF), frekuensi tinggi (*high frequency*, HF), frekuensi ultra tinggi (*ultrahigh frequency*, UHF), atau gelombang mikro (*microwave*). Contoh aplikasi *tag* pasif adalah pada sistem angkutan massal (*Mass Rapid Transit* - MRT), autentikasi masuk gedung dan barang-barang konsumsi. Harga *tag* pasif lebih murah dibandingkan harga *tag* versi lainnya. Perkembangan *tag* murah ini telah menciptakan revolusi dalam pengadopsian RFID dan memungkinkan penggunaannya dalam skala yang luas baik oleh organisasi-organisasi pemerintah maupun industri.

b. *Tag* Semipasif

Tag semipasif adalah versi *tag* yang memiliki catu daya sendiri (baterai) tetapi tidak dapat menginisiasi komunikasi dengan *reader*. Dalam hal ini baterai digunakan oleh *tag* sebagai catu daya untuk melakukan fungsi lain seperti pemantauan keadaan lingkungan dan mencatu bagian elektronik internal *tag* serta untuk memfasilitasi penyimpanan informasi. *Tag* versi ini tidak secara aktif memancarkan sinyal ke *reader*. Sebagian *tag* semipasif tetap dalam keadaan siap (*stand by*) hingga menerima sinyal dari *reader*. *Tag* semipasif dapat dihubungkan dengan sensor untuk menyimpan informasi pada peralatan keamanan *container*.

c. *Tag* Aktif

RFID *tag* yang aktif, di sisi lain harus memiliki *power supply* sendiri dan memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung berbagai macam informasi didalamnya. Jarak jangkauan dari *tag* yang aktif ini bisa sampai sekitar 100 meter.

Tabel 2.1 Karakteristik *Tag* RFID

Kriteria	<i>Tag</i> Aktif	<i>Tag</i> Semipasif	<i>Tag</i> Pasif
Catu Daya	Baterai Internal	Baterai Internal	Eksternal (dari <i>reader</i>)
Tipe Memori	<i>Read-Write</i>	<i>Read Write</i>	<i>Read Only</i>
Rentang Baca	Dapat Mencapai 750 kaki	Dapat Mencapai 100 kaki	Dapat Mencapai 20 kaki
Usia <i>Tag</i>	5 - 10 Tahun	2 – 7 Tahun	> 20 Tahun

2.1.2 *Reader* RFID

Sebuah *reader* RFID menggunakan antenanya sendiri untuk berkomunikasi dengan *tag*. Ketika *reader* memancarkan gelombang radio, seluruh *tag* yang dirancang pada frekuensi tersebut serta berada pada rentang bacanya akan memberikan respon. Sebuah *reader* juga dapat berkomunikasi dengan *tag* tanpa *line of sight* langsung, tergantung kepada frekuensi radio dan tipe *tag* (aktif, pasif atau semipasif) yang digunakan.

Reader dapat memproses banyak item sekaligus. Menurut bentuknya, *reader* dapat berupa *reader* bergerak seperti peralatan genggam, atau stasioner seperti peralatan *point-of-sale* di supermarket. *Reader* dibedakan berdasarkan kapasitas penyimpanannya, kemampuan pemrosesannya, serta frekuensi yang dapat dibacanya. Basis data merupakan sebuah sistem informasi logistik pada posisi *back-end* yang bekerja melacak dan menyimpan informasi tentang item bertag. Informasi yang tersimpan dalam basis data dapat terdiri dari *identifier item*, deskripsi, pembuat, pergerakan dan lokasinya. Tipe informasi yang disimpan dalam basis data dapat bervariasi tergantung kepada aplikasinya. Sebagai contoh, data yang disimpan pada sistem pembayaran tol akan berbeda dengan yang disimpan pada rantai *supply*.

2.2 Sistem dan Cara Kerja RFID [2]

Cara kerja RFID apabila pembaca RFID diberi tegangan dan arus yang mengalir melalui lilitan antena akan membangkitkan *fluks* magnetik. Hal tersebut menyebabkan pembaca RFID memancarkan frekuensi radio secara terus menerus. Apabila *tag* berada dalam jangkauan pembaca maka *chip* yang ada pada RFID akan dibangkitkan

melalui tegangan terinduktansi. Setelah *chip* dibangkitkan maka *tag* RFID akan mengirimkan nomor unik yang tersimpan ke pembaca RFID untuk di baca. Setelah itu pembaca RFID akan meneruskan data yang dibaca ke komputer. Kegunaan dari sistem RFID ini adalah untuk mengirimkan data dari piranti *portable*, yang dinamakan *tag*, dan kemudian dibaca oleh RFID *reader* dan kemudian diproses oleh aplikasi komputer yang membutuhkannya. Data yang dipancarkan dan dikirimkan tadi bisa berisi beragam informasi, seperti ID, informasi lokasi atau informasi lainnya seperti harga, warna, tanggal pembelian dan lain sebagainya. Gambar 2.2 merupakan sistem kerja dari RFID



Gambar 2.2 Sistem Kerja RFID

Dalam suatu sistem RFID sederhana, suatu obyek dilengkapi dengan *tag* yang kecil dan murah. *Tag* tersebut berisi *transponder* dengan suatu *chip* memori digital yang di dalamnya berisi sebuah kode produk yang sifatnya unik. Sebaliknya, *interrogator* suatu antenna yang berisi *transceiver* dan *decoder*, memancarkan sinyal yang bisa mengaktifkan RFID *tag* sehingga dia dapat membaca dan menulis data ke dalamnya. Ketika suatu RFID *tag* melewati suatu zone elektromagnetis, maka dia akan mendeteksi sinyal aktivasi yang dipancarkan oleh *reader*. *Reader* akan men-*decode* data yang ada pada *tag* dan kemudian data tadi akan diproses oleh komputer. Berdasarkan kemampuan dibaca dan ditulis RFID dikelompokkan sebagai *read only* label yang berisi nomor unik yang tidak dapat berubah, *Worm Write Once Read Many* memungkinkan untuk mengkodekan mengisi untuk pertama kali dan kemudian data atau kode tersebut terkunci dan tidak dapat dirubah dan *Read/Write* Dimungkinkan untuk mengisi dan memperbarui informasi didalamnya.

Pemilihan frekuensi radio merupakan kunci karakteristik operasi sistem RFID. Frekuensi sebagian besar ditentukan oleh kecepatan komunikasi dan jarak baca terhadap *tag*. Secara umum tingginya frekuensi mengindikasikan jauhnya jarak baca. Frekuensi yang lebih tinggi mengindikasikan jarak baca yang lebih jauh. Pemilihan tipe frekuensi juga dapat ditentukan oleh tipe aplikasinya. Aplikasi tertentu lebih cocok untuk salah satu tipe frekuensi dibandingkan dengan tipe lainnya karena gelombang radio memiliki perilaku yang berbeda-beda menurut frekuensinya.

Ada beberapa *band* frekuensi yang digunakan untuk sistem RFID, antara lain:

a. *Band LF (Low Frequency)*

Beroperasi pada kisaran dari 125 KHz hingga 134 KHz. *Band* ini paling sesuai untuk penggunaan jarak pendek (*short-range*) seperti sistem antipencurian, identifikasi hewan dan sistem kunci mobil.

b. *Band HF (High Frequency)*

Beroperasi di sekitar 900 MHz dan dapat dibaca dari jarak yang lebih jauh dari *tag* HF, berkisar dari 3 hingga 15 kaki. *Tag* ini lebih sensitif terhadap faktor-faktor lingkungan daripada *tag-tag* yang beroperasi pada frekuensi lainnya. *Band* 900 MHz muncul sebagai *band* yang lebih disukai untuk aplikasi rantai *supply* disebabkan laju dan rentang bacanya. *Tag* UHF pasif dapat dibaca dengan laju sekitar 100 hingga 1.000 *tag* perdetik. *Tag* ini umumnya digunakan pada pelacakan kontainer, truk, trailer, dan terminal peti kemas.

c. Gelombang Mikro

Tag yang beroperasi pada frekuensi gelombang mikro, biasanya 2,45 dan 5,8 Gigahertz (GHz), mengalami lebih banyak pantulan gelombang radio dari obyek-obyek di dekatnya yang dapat mengganggu kemampuan *reader* untuk berkomunikasi dengan *tag*. *Tag* RFID gelombang mikro biasanya digunakan untuk manajemen rantai *supply*.

2.3 *Push Button* [3]

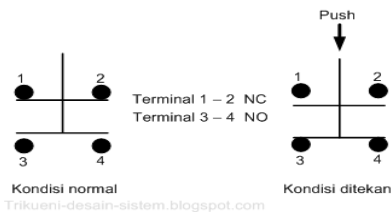
Push button (saklar tombol tekan) adalah perangkat/saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai *device*

penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal. Gambar 2.3 merupakan bentuk fisik dari *push button*.



Gambar 2.3 *Push Button*

Sebagai *device* penghubung atau pemutus, *push button switch* hanya memiliki 2 kondisi, yaitu *On* dan *Off* (1 dan 0). Istilah *On* dan *Off* ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi *On* dan *Off*. Karena sistem kerjanya yang *unlock* dan langsung berhubungan dengan operator, *push button switch* menjadi *device* paling utama yang biasa digunakan untuk memulai dan mengakhiri kerja mesin di industri. Secanggih apapun sebuah mesin bisa dipastikan sistem kerjanya tidak terlepas dari keberadaan sebuah saklar seperti *push button switch* atau perangkat lain yang sejenis yang bekerja mengatur pengkondisian *On* dan *Off*. Gambar 2.4 merupakan sketsa prinsip kerja *push button*.



Gambar 2.4 Prinsip Kerja *Push Button*

Berdasarkan fungsi kerjanya yang menghubungkan dan memutuskan, *push button switch* mempunyai 2 tipe kontak yaitu NC (*Normally Close*) dan NO (*Normally Open*).

- a. NO (*Normally Open*), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir). Dan ketika tombol saklar ditekan, kontak yang NO ini akan menjadi menutup (*Close*) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan sistem *circuit* (*Push Button ON*).

- b. NC (*Normally Close*), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus listrik). Dan ketika tombol saklar *push button* ditekan, kontak NC ini akan menjadi membuka (*Open*), sehingga memutus aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan sistem *circuit* (*Push Button Off*).

2.4 Buzzer [3]

Buzzer adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Suara yang dihasilkan oleh *buzzer* tergolong sederhana yakni monofonik, berbeda dengan *speaker* yang mengeluarkan suara yang lebih variatif (polifonik). Jenis *buzzer* yang sering ditemukan dan digunakan adalah *buzzer* yang berjenis *piezoelectric*. *Buzzer* yang termasuk dalam keluarga transduser ini juga sering disebut dengan *beeper* seperti yang ditunjukkan pada Gambar 2.5.



Gambar 2.5 Bentuk Fisik *Buzzer*

Piezoelectric buzzer adalah jenis *buzzer* yang menggunakan efek *piezoelectric* untuk menghasilkan suara atau bunyinya. Tegangan listrik yang diberikan ke bahan *piezoelectric* akan menyebabkan gerakan mekanis, gerakan tersebut kemudian diubah menjadi suara atau bunyi yang dapat didengar oleh telinga manusia dengan menggunakan diafragma dan resonator. *Buzzer piezoelectric* dapat bekerja dengan baik dalam menghasilkan frekuensi di kisaran 1 – 5 kHz hingga 100 kHz untuk aplikasi *ultrasound*. Tegangan operasional *buzzer piezoelectric* yang umum biasanya berkisar diantara 3 Volt hingga 12 Volt.

2.5 LCD 16x4 [3]

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada

di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. Gambar 2.6 merupakan tampilan dari LCD.



Gambar 2.6 Tampilan LCD 16x4

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan *sandwich* memiliki *polarizer* cahaya vertikal depan dan *polarizer* cahaya horisontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.

Dalam modul LCD (*Liquid Cristal Display*) terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD (*Liquid Cristal Display*). Mikrokontroler pada suatu LCD (*Liquid Cristal Display*) dilengkapi dengan memori dan *register*. Memori yang digunakan mikrokontroler internal LCD adalah:

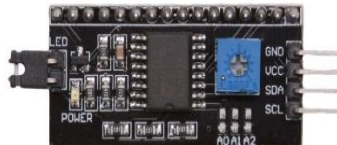
1. DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada.
2. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
3. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD (*Liquid Cristal*

Display) tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.

4. *Register* perintah yaitu *register* yang berisi perintah-perintah dari mikrokontroler ke panel LCD (*Liquid Cristal Display*) pada saat proses penulisan data atau tempat status dari panel LCD (*Liquid Cristal Display*) dapat dibaca pada saat pembacaan data.
5. *Register* data yaitu *register* untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada *register* akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

2.6 Modul I2C LCD [4]

Modul I2C LCD adalah modul LCD yang dikendalikan secara serial sinkron dengan protokol I2C. Normalnya, modul LCD dikendalikan secara paralel baik untuk jalur data maupun kontrolnya. Namun, jalur paralel akan memakan banyak pin di sisi kontroler. Akan dibutuhkan setidaknya 6 atau 7 pin untuk mengendalikan sebuah modul LCD. Dengan demikian untuk sebuah kontroler yang ‘sibuk’ dan harus mengendalikan banyak I/O, menggunakan jalur paralel adalah solusi yang kurang tepat. Gambar 2.7 merupakan bentuk fisik modul I2C LCD.



Gambar 2.7 Modul I2C LCD

Sebagai contoh, sebuah *Arduino Uno* memiliki pin digital sebanyak 13 buah. Jika separuhnya digunakan untuk mengendalikan LCD berarti pada *Arduino* hanya tersisa alternatif sekitar 6 atau 7 pin untuk mengendalikan perangkat lain, misalnya motor DC, sensor ultrasonik, keypad, dsb. Jika alternatif 6 atau 7 pin tersebut tidak cukup untuk mengendalikan perangkat lain, jalur kendali LCD dapat diubah dari paralel ke serial menggunakan modul I2C untuk LCD, sehingga hanya membutuhkan 2 jalur kabel saja (plus *ground* dan VCC).

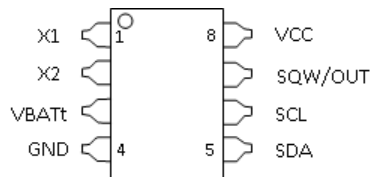
2.7 RTC (*Real Time Clock*) [4]

Real-Time Clock atau disingkat RTC adalah jam elektronik berupa *chip* yang dapat menghitung waktu ,mulai dari detik, menit, jam hingga tahun dengan akurat dan menjaga atau menyimpan data waktu tersebut secara *real time*. Karena jam tersebut bekerja *real time*, maka setelah proses hitung waktu dilakukan *output* datanya langsung disimpan atau dikirim ke *device* lain melalui sistem antarmuka.

Chip RTC sering dijumpai pada *motherboard* PC (biasanya terletak dekat *chip* BIOS). Semua komputer menggunakan RTC karena berfungsi menyimpan informasi jam terkini dari komputer yang bersangkutan. RTC juga biasanya digunakan pada mikrokontroler. RTC dilengkapi dengan baterai sebagai pensuplai daya pada *chip*, sehingga jam akan tetap *up-to-date* walaupun komputer atau mikrokontroler dimatikan. RTC dinilai cukup akurat sebagai pewaktu (*timer*) karena menggunakan osilator kristal.

2.7.1 RTC DS1307

RTC tipe DS1307 merupakan jenis pewaktu yang menggunakan komunikasi serial untuk operasi tulis baca, dengan spesifikasi RTC meyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100, 56-byte, *battery-backed*, RAM *nonvolatile* (NV) RAM untuk penyimpanan, antarmuka serial *Two-wire* (I2C), sinyal luaran gelombang-kotak terprogram (*programmable squarewave*), deteksi otomatis kegagalan-daya (*power-fail*) dan rangkaian *switch*, konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional osilator, tersedia fitur industri dengan ketahanan suhu: -40°C hingga +85°C, tersedia dalam kemasan 8-pin DIP atau SOI pada Gambar 2.8 merupakan konfigurasi pin DS1307 dan Tabel 2.2 adalah fungsi pin I/O DS1307.



Gambar 2.8 Konfigurasi Pin DS1307

Tabel 2.2 Fungsi Pin I/O DS1307

No	Nama Pin	Fungsi
1	X1	<i>Oscillator Crystal</i>
2	X2	32,768KHz
3	VBAT	Battery Input (+3V)
4	GND	<i>Ground</i>
5	SDA	Serial Data
6	SCL	<i>Serial Clock Input</i>
7	SQW/OUT	<i>Square Wave/Output Driver</i>
8	VCC	<i>Supply Power</i>

Pin-pin yang terdapat pada RTC DS1307 memiliki berbagai fungsi seperti berikut ini :

- X1 dan X2 adalah pembangkit pulsa yang terhubung dengan *quart* kristal 32,768 kHz sebagai *internal circuit oscillator* yang didesain dengan C_L 12,5pF. X1 adalah osilator *input* yang terhubung langsung dengan kristal, sedang X2 adalah keluaran dari osilator kristal internal.
- *Power Control* merupakan catu daya yang mensuplai tegangan ke DS1307, Vcc adalah daya luar sebesar 5V dan V_{BAT} sebagai suplai *input* dengan 3V *lithium Cell*. Tegangan baterai harus berada diantara batasan minimum dan maksimum pengoperasian. Baterai *lithium* 48mAh atau lebih dapat mempertahankan fungsi RTC selama 10 tahun.
- *Serial Bus Interface And Address Register* merupakan jalur data serial dan pengalamatan *register* DS1307 dengan akses pulsa melalui SCL dan SDA. SCL (*Serial Clock*) berfungsi sebagai *clock input* I²C dan digunakan untuk sinkronisasi data serial. SDA (*Serial Data*) berfungsi sebagai data *input/output* untuk I²C serial, baik SCL dan SDA masih memerlukan *pull up* eksternal.
- *Control Logic* merupakan pengendali data-data yang dibaca ataupun ditulis melalui SCL dan SDA dengan pewaktu dari osilator kristal.

- *Buffer (7 bytes)* adalah penyangga sementara sebelum data diterima atau dikirim, berkisar 7 bytes (7 x 8 bit) sebagai transit pengalaman *register* 8 bit detik-menit-jam-hari-tanggal-bulan-tahun.
- *Clock, Calender and Control Register* atau CR berisi informasi *clock* dan kalender serta *register* pengendali untuk mengontrol pengoperasian SQW/OUT. *Clock and Calender* merupakan *register* RTC dengan waktu dan kalender dengan akses pembacaan dan penulisan *register byte* dalam format BCD.

2.8 Mifare RC522 RFID Reader Module [5]

Mifare RC522 RFID Reader Module adalah sebuah modul berbasis IC Philips MFRC522 yang dapat membaca RFID dengan penggunaan yang mudah dan harga yang murah, karena modul ini sudah berisi komponen-komponen yang diperlukan oleh MFRC522 untuk dapat bekerja. Modul ini dapat digunakan langsung oleh MCU dengan menggunakan *interface* SPI, dengan suplai tegangan sebesar 3,3V. MFRC522 merupakan produk dari NXP yang menggunakan *fully integrated* 13,56MHz *non-contact communication card* chip untuk melakukan pembacaan maupun penulisan. MFRC522 support dengan semua varian MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 dan MIFARE Plus RF *identification protocols*.

Dalam hal kecepatan baca, RFID reader mampu membaca data dari kartu sebesar 16 *byte* dalam waktu rata-rata selama 9,5 ms, untuk kecepatan tulis didapatkan waktu rata-rata selama 10 ms. Sementara dalam hal kecepatan transfer data dari *Arduino* ke database, waktu rata-rata pengiriman adalah sebesar 7 ms. Hal ini menunjukkan, bahwa sistem yang dibuat mampu mengirimkan data secara cepat dan tepat. Gambar 2.9 merupakan tampilan modul reader MFRC522 dan Tabel 2.3 merupakan spesifikasi MFRC522



Gambar 2.9 Tampilan Modul *Reader* MRC522

Tabel 2.3 Spesifikasi MRC522

Spesifikasi	Keterangan
<i>Chipset</i>	MFRC522 <i>Contactless Reader/Writer</i>
Frekuensi	13,56 MHz
Jarak pembacaan Kartu	< 50mm
Protokol Akses	SPI (Serial Peripheral <i>Interface</i>) 10 Mbps
Kecepatan Transmisi RF	424 kbps (dua arah / <i>bi-directional</i>) / 848 kbps (<i>unidirectional</i>)
Catu Daya	3,3 Volt
Konsumsi Arus	13-26 mA pada saat operasi baca/tulis, < 80 μ A saat modus siaga
Suhu Operasional	-20°C s.d. +80°C
Dimensi	40 x 50 mm

2.9 MP3 *Shield* [6]

Modul MP3 *shield* untuk arduino ini adalah sebuah modul yang mampu membuat *Arduino* memutar *file* MP3 dengan perintah dari koding yang telah dibuat pada *Arduino*. *File* pada MP3 akan diputar sesuai dengan kondisi koding yang ada. Dengan begitu dapat membuat beberapa *project* mikrokontroler yang *output*-nya berupa *file* MP3 sebagai reaksi dari fungsi yang telah dilakukan. Beberapa *project* yang bisa dibuat dengan modul MP3 *shield* diantaranya jam berbicara, alarm jadwal sholat, bel sekolah otomatis, mesin antrian sederhana dan masih banyak lagi.

MP3 *Shield* ini adalah revisi terbaru dari MP3 *Player Shield*. *Shield* ini masih mempertahankan kemampuan dari MP3 dekode-nya,

akan tetapi *shield* ini ditambahkan fungsionalnya, dengan menambahkan slot kartu *MicroSD*. Oleh Karena itu dapat mengambil *file* MP3 dari kartu *MicroSD* hanya dengan menggunakan Satu *Shield*. Dan sangat efektif untuk mengubah *Arduino* apapun menjadi MP3 *player*. MP3 *shield* ini masih memanfaatkan IC VS1053B audio MP3 *decoder*. IC ini fungsinya untuk memecahkan kode *file audio*. Gambar 2.10 merupakan bentuk fisik modul MP3 *shield* dan Tabel 2.4 merupakan karakteristik dari modul MP3 *shield*.



Gambar 2.10 Bentuk Fisik Modul MP3 *Shield*

VS1053 menerima *file input* melalui *bus input serial* (SPI). Setelah *file* itu diterjemahkan oleh IC *audio* tersebut dikirim keluar ke *stereo headphone* yang menggunakan *jack* 3,5mm, serta 2-pin 0,1" *Pitch header*. VS1053 adalah sebuah IC serbaguna "MP3 *decoder chip*" milik golongan prosesor *audio VLSI*. Selain mampu men-*decode* semua format yang paling umum *audio* termasuk fitur-fitur canggih baru *file AAC* - fungsi IC ini dapat sangat diperluas hanya dengan memuat sedikit tambahan *software* untuk memori RAM. Selain mampu men-*decode* semua format *audio* utama VS1053 mampu merekam dalam tiga format *audio* yang berbeda.

Tabel 2.4 Karakteristik Modul MP3 *Shield*

No.	Features
1.	<i>Plug in and Play for Arduino</i>
2.	<i>Control by serial UART interface or keys onboard</i>
3.	<i>Directly drive 32ohm headset</i>
4.	<i>On-board Audio Amplifier to drive 3W/4Ω external loudspeaker</i>
5.	<i>32 Levels adjustable volume</i>
6.	<i>Support FAT16/FAT32 file system</i>
7.	<i>Support up to SD/SDHC card and U-Disk</i>

No.	Features
8.	On board 64Mbit SPI-FLASH
9.	Support copying audio files from SD Ccrd or U-Disk to SPI-FLASH
10	Supporting sleep mode
11.	Support 16k-320Kbps MP3 files and 8k-44.1KHz sampling rate WAV files (SPI-FLASH only supports MP3 formats)

2.10 Arduino Mega 2560 [7]

Arduino Mega 2560 adalah papan mikrokontroler berbasis ATmega 2560. *Arduino Mega 2560* memiliki 54 pin *digital input / output*, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, dan 4 pin sebagai UART (*port serial hardware*), 16 MHz kristal osilator, koneksi USB, *jack power*, *header ICSP*, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau *power* dihubungkan dengan *adaptor* AC – DC atau baterai untuk mulai mengaktifkannya. *Arduino Mega 2560* kompatibel dengan sebagian besar *shield* yang dirancang untuk *Arduino Duemilanove* atau *Arduino Diecimila*. *Arduino Mega 2560* adalah versi terbaru yang menggantikan versi *Arduino Mega*. Gambar 2.11 merupakan bentuk fisik dari *Arduino mega 2560* dan pada Tabel 2.5 adalah spesifikasi *Arduino mega 2560*.



Gambar 2.11 Bentuk Fisik *Arduino Mega 2560*

Tabel 2.5 Spesifikasi *Arduino Mega 2560*

Spesifikasi	Keterangan
Chip mikrokontroler	ATmega 2560
Tegangan operasi	5V

Spesifikasi	Keterangan
Tegangan <i>input</i> (<i>jack</i> DC)	7V – 12V
Tegangan <i>input</i> (<i>limit</i> , <i>jack</i> DC)	6V – 20V
Digital I/O pin	54 buah, 6 diantaranya menyediakan PWM <i>output</i>
Analog <i>Input</i> pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3,3V	50 mA
Memori <i>Flash</i>	256 KB, 8 KB telah digunakan untuk <i>bootloader</i>
SRAM	8 Kb
EEPROM	4 Kb
<i>Clock speed</i>	16Mhz
Dimensi	101,5 mm x 53,4 mm
Berat	37 g

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (*non-USB*) dapat berasal baik dari adaptor AC-DC atau baterai. *Adaptor* dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin Gnd dan pin Vin dari konektor *power*.

Papan *Arduino Mega* 2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 Volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt.

Pin tegangan yang tersedia pada papan *Arduino* adalah sebagai berikut:

- VIN : Adalah *input* tegangan untuk papan *Arduino* ketika menggunakan sumber daya eksternal (sebagai 'saingan' tegangan 5 Volt dari koneksi USB atau sumber daya ter-regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui *jack power*, kita bisa mengakses/mengambil tegangan melalui pin ini.
- 5V : Sebuah pin yang mengeluarkan tegangan ter-regulator 5 Volt, dari pin ini tegangan sudah diatur (ter-regulator) dari regulator yang tersedia (*built-in*) pada papan. *Arduino* dapat diaktifkan dengan sumber daya baik berasal dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada board (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator dapat merusak papan *Arduino*.
- 3V3 : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (*on-board*). Arus maksimum yang dihasilkan adalah 50 mA.
- GND : Pin *Ground* atau Massa.
- IOREF : Pin ini pada papan *Arduino* berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (*voltage translator*) pada *output* untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

Arduino Mega 2560 memiliki 256 KB *flash memory* untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Masing-masing dari 54 digital pin pada *Arduino Mega* dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. *Arduino mega* beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara *default*) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- Serial : 0 (RX) dan 1 (TX); Serial 1 : 19 (RX) dan 18 (TX); Serial 2 : 17 (RX) dan 16 (TX); Serial 3 : 15 (RX) dan 14 (TX).

Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin *chip* ATmega16U2 Serial USB-*to*-TTL.

- Eksternal Interupsi : Pin 2 (*interrupt* 0), pin 3 (*interrupt* 1), pin 18 (*interrupt* 5), pin 19 (*interrupt* 4), pin 20 (*interrupt* 3), dan pin 21 (*interrupt* 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.
- SPI : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan *header* ICSP, yang secara fisik kompatibel dengan *Arduino Uno*, *Arduino Duemilanove* dan *Arduino Diecimila*.
- LED : Pin 13. Tersedia secara *built-in* pada papan arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala (ON), dan ketika pin diset bernilai *LOW*, maka LED padam (*off*).
- TWI : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan *Wire*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada *Arduino Duemilanove* atau *Arduino Diecimila*.

Arduino Mega 2560 memiliki 16 pin sebagai analog *input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analogReference()*. Ada beberapa pin lainnya yang tersedia, antara lain:

- AREF : Referensi tegangan untuk *input* analog. Digunakan dengan fungsi *analogReference()*.
- RESET : Jalur *LOW* ini digunakan untuk me-*reset* (menghidupkan ulang) mikrokontroler.

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan *Arduino* lain, atau dengan mikrokontroler lainnya. *Arduino* ATmega328 menyediakan 4 *hardware* komunikasi serial UART TTL (5 Volt). Sebuah *chip* ATmega16U2 (ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai COM *port virtual* (pada *device* komputer) untuk

berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file *inf*, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai *port* COM secara otomatis. Perangkat lunak *Arduino* termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan *Arduino*. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip* USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

Sebuah perpustakaan *SoftwareSerial* memungkinkan untuk komunikasi serial pada salah satu pin digital *Mega* 2560. ATmega2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak *Arduino* termasuk perpustakaan *Wire* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan perpustakaan SPI.

Arduino Mega 2560 memiliki *polyfuse reset* yang melindungi *port* USB komputer Anda dari hubungan singkat dan arus lebih. Meskipun pada dasarnya komputer telah memiliki perlindungan internal pada *port* USB mereka sendiri, sekring memberikan lapisan perlindungan tambahan. Jika arus lebih dari 500 mA dihubungkan ke *port* USB, sekring secara otomatis akan memutuskan sambungan sampai hubungan singkat atau *overload* dihapus/dibuang.

Maksimum panjang dan lebar PCB *Mega* 2560 adalah 4 x 2,1 inch (10,16 x 5,3 cm), dengan konektor USB dan *jack power* menonjol melampaui batas dimensi. Empat lubang sekrup memungkinkan papan terpasang pada suatu permukaan atau wadah. Perhatikan bahwa jarak antara pin digital 7 dan 8 adalah 160 mil (0,16"), tidak seperti pin lainnya dengan kelipatan genap berjarak 100 mil. *Arduino Mega* 2560 dirancang agar kompatibel dengan sebagian shield yang dirancang untuk *Arduino Uno*, *Arduino Diecimila* atau *Arduino Duemilanove*. Pin Digital 0-13 (pin AREF berdekatan dan pin GND), *input* analog 0 sampai 5, *header power*, dan *header ICSP* berada di lokasi yang ekuivalen. Selanjutnya UART utama (*port* serial) terletak di pin yang sama (0 dan 1), seperti pin interupsi eksternal 0 dan 1 (masing-masing pada pin 2 dan 3). SPI di kedua *header ICSP* yaitu *Mega* 2560 dan *Duemilanove/Diecimila*. Pin I2C tidak terletak pada pin yang sama pada *Mega* pin (20 dan pin 21) seperti halnya *Duemilanove/Diecimila* (*input* analog pin 4 dan pin 5).

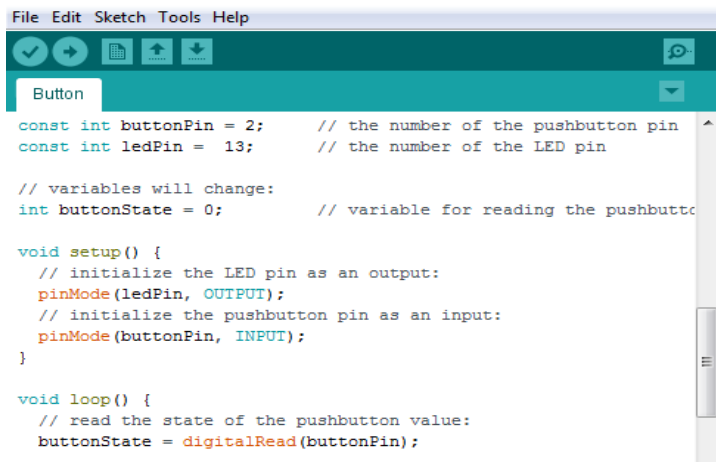
2.10.1 Pemrograman Arduino

Integrated Development Environment (IDE) *Arduino* merupakan aplikasi yang mencakup *editor*, *compiler*, dan *uploader* dapat menggunakan semua seri modul keluarga *Arduino*, seperti *Arduino Duemilanove*, *Uno*, *Bluetooth*, *Mega*. Kecuali ada beberapa tipe *board* produksi *Arduino* yang memakai *microcontroller* di luar seri AVR, seperti mikroprosesor ARM. Saat menulis kode program atau mengkompilasi modul *hardware Arduino* tidak harus tersambung ke PC atau *Notebook*, walaupun saat proses unggahan ke *board* diperlukan modul *hardware*.

IDE *Arduino* juga memiliki keterbatasan tidak mendukung fungsi *debugging hardware* maupun *software*. Proses kompilasi IDE *Arduino* diawali dengan proses pengecekan kesalahan sintaksis *sketch*, kemudian memanfaatkan pustaka *Processing* dan *avr – gcc sketch* dikompilasi menjadi berkas objek, lalu berkas-berkas objek digabungkan oleh pustaka *Arduino* menjadi berkas biner. Berkas biner ini diunggah ke *chip microcontroller* via kabel USB, serial *port* DB9, atau Serial *Bluetooth*.

Compiler IDE *Arduino* juga memanfaatkan pustaka *open source* *AVRLibc* sebagai standar *de-facto* pustaka referensi dan fungsi *register microcontroller* AVR. Pustaka *AVRLibc* ini sudah disertakan dalam satu paket program IDE *Arduino*. Meskipun demikian, kita tidak perlu mendefinisikan *directive#include* dari pustaka *AVRLibc* pada *sketch* karena otomatis *compiler* me-link pustaka *AVRLibc* tersebut.

Ukuran berkas biner HEX hasil kompilasi akan semakin besar jika kode *sketch* semakin kompleks. Berkas biner memiliki ekstensi hex berisi data instruksi program yang biasa dipahami oleh *microcontroller* target. Selain itu, *port* paralel juga bias dipakai untuk mengunggah *bootloader* ke *microcontroller*. Meskipun demikian, cara ini sudah jarang digunakan karena sekarang hampir tidak ada *mainboard* PC yang masih menyediakan *port* paralel, dan pada *notebook* juga sudah tidak menyertakan *port* paralel. Gambar 2.12 merupakan contoh pemrograman pada *Arduino*.



Gambar 2.12 Contoh Pemrograman Arduino

2.11 Visual Basic 6.0 [8]

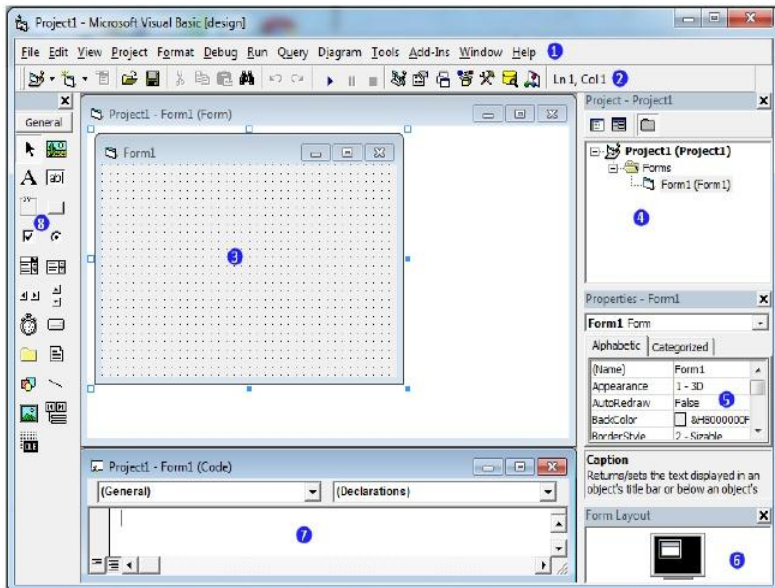
Visual Basic 6.0 adalah bahasa pemrograman berbasis *Windows*. Saat ini, *Visual Basic* adalah salah satu bahasa pemrograman yang terbaik. *Visual Basic* merupakan pengembangan dari *Basic*. *Basic* (*Beginner's All-purpose Symbolic Instruction Code*) adalah sebuah bahasa pemrograman “kuno” yang merupakan awal dari bahasa-bahasa pemrograman tingkat tinggi lainnya yang mana bahasa yang digunakan kurang familiar dengan bahasa manusia. *Basic* dirancang pada tahun 1950-an dan ditujukan untuk dapat digunakan oleh para *programmer* pemula. *Visual Basic* ini memiliki bahasa pemrograman yang lebih familiar dan mudah dimengerti oleh manusia.

Visual Basic atau biasa disebut VB merupakan sebuah bahasa pemrograman *event driven* yang artinya program menunggu sampai adanya respon dari pemakai berupa *event*/kejadian tertentu. Ketika *event* terdeteksi, kode yang berhubungan dengan *event* (prosedur *event*) akan dijalankan (*driven*). *Visual Basic* juga merupakan bahasa pemrograman *Integrated Development Environment* (IDE) yang artinya program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. *Visual Basic* ini dikembangkan oleh *Microsoft* yang bertujuan untuk membuat program atau perangkat lunak berbasis

Windows dengan menggunakan model pemrograman COM (*Component Object Model*).

Visual Basic merupakan salah satu bahasa pemrograman komputer yang berorientasi *object* (*Object Oriented Programming* = OOP) yaitu bahasa pemrograman yang mana semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar.

Lingkungan *Visual Basic* atau sering disebut juga *Integrated Development Environment* (IDE) VB 6 merupakan halaman kerja tempat program itu dibuat baik itu *interfacenya* maupun bahasa pemrogramannya. Tentunya untuk mengetahui IDE VB 6 ini tentunya kita harus mempunyai *software* yang sudah siap kita jalankan pada PC kita.



Gambar 2.13 Tampilan Lingkungan Kerja *Visual Basic* 6.0

Lingkungan kerja VB 6 memiliki beberapa bagian-bagian yang memiliki fungsi masing-masing, seperti pada Gambar 2.13, yaitu:

1. *Menubar*

Menubar dalam VB seperti yang biasa kita lihat dalam MS Office. Di dalamnya terdapat menu *File, Edit, View, Project, Format*, dan sebagainya.

2. *Toolbar*

Toolbar adalah sebuah *window* yang berisi ikon-ikon yang fungsinya sama dengan menu tetapi dapat digunakan dengan lebih cepat karena sebuah ikon mewakili satu perintah tertentu. Contohnya ikon *open, save, copy*, dan sejenisnya.

3. *Jendela Form*

Form adalah sebuah objek yang digunakan untuk menempatkan objek-objek dari *toolbox*.

4. *Project Explorer*

Satu *project* aplikasi biasanya terdiri dari beberapa *form* dan komponen lainnya, untuk mempermudah berpindah antara komponen dalam *Visual Basic* dapat menggunakan jendela *project*. Untuk berpindah dari *form1* ke *form2* cukup dengan mengklik tulisan *form2* pada *project explorer*.

5. *Jendela Properties*

Fungsi dari jendela *properties* adalah untuk mengubah *setting property* dari setiap elemen yang terdapat pada *form*.

6. *Form Layout*

Form layout merupakan gambaran tampilan dari program yang kita lihat jika sudah di *export* menjadi *file .exe*.

7. *Jendela Code*

Jendela *code* merupakan tempat kita menuliskan *code* perintah untuk menjalankan suatu proses.

8. *Toolbox*

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu *project* aktif, artinya isi komponen dalam *toolbox* sangat tergantung pada jenis *project* yang dibangun.

2.12 Obyek Museum [9]

Obyek museum merupakan obyek yang terdapat pada museum untuk dipajang agar dapat dilihat oleh pengunjung dan merupakan barang-barang koleksi museum. Berikut adalah beberapa contoh obyek yang terdapat pada museum:

2.12.1 *Map of Hell*

Mappa dell'Inferno (*Map of Hell*) karya dari Sandro Botticelli, biasa disebut *The Abyss of Hell* atau *La voragine dell'Inferno*, adalah salah satu perkamen yang didesain oleh pelukis Italia tersebut untuk mengilustrasikan sebuah edisi dari *The Divine Comedy* karya Dante Alighieri. Perkamen Peta Neraka menunjukkan geografi dari neraka pada bagian corong klasik. Perkamen tersebut dilukis oleh Botticelli antara tahun 1480 dan 1490. Struktur dari neraka dibuat oleh Dante dan didesain oleh Botticelli dengan bentuk kerucut terbalik, seperti corong, dengan 9 tingkat lingkaran yang menurun ke inti bumi, dimana Lucifer berada. Pada setiap lingkaran, para pendosa dihukum menurut dosanya, dengan struktur seperti berikut: Lingkaran pertama untuk *Limbo* (orang-orang yang tidak dibaptis dan para penyembah berhala), lingkaran kedua untuk nafsu, lingkaran kedua untuk kerakusan, lingkaran keempat untuk ketamakan, lingkaran kelima untuk untuk kemarahan dan kemalasan, lingkaran keenam untuk klenik, lingkaran ketujuh untuk kekerasan, lingkaran kedelapan untuk penipuan, dan lingkaran kesembilan untuk pengkhianatan. Gambar 2.14 merupakan gambar lukisan *Map of Hell*.



Gambar 2.14 Lukisan *Map of Hell*

2.12.2 *The Scream*

The Scream adalah nama populer yang diberikan untuk empat versi dari sebuah komposisi, yang dibuat dalam bentuk lukisan cat dan pastel, oleh seniman ekspresionis asal Norwegia yang bernama Edvard Munch antara tahun 1893 dan 1910. Karyanya menunjukkan sebuah gambar dengan sebuah ekspresi penderitaan terhadap pemandangan dengan langit jingga yang riuh. Karya ini merupakan salah satu dari dua versi lukisan cat yang dilukis pada 1893. Pada buku hariannya dengan judul "Nice 22 January 1892", Munch menulis:

Aku berjalan melintasi jalan dengan dua teman – matahari terbenam – tiba-tiba langit menjadi semerah darah – aku berhenti, merasa lelah, dan bersandar pada pagar – terdapat darah dan lidah api diatas *fjord* yang berwarna biru kehitaman dan di kota – teman-temanku berjalan, dan aku berdiri di sana gemetar dengan kegelisahan – dan aku merasakan teriakan tak berujung melalui alam.

Dia lalu mendeskripsikan inspirasinya untuk gambarnya:

Di suatu malam aku berjalan di sepanjang jalan kecil, dengan kota di satu sisi dan *fjord* di bawahnya. Aku merasa lelah dan sakit. Aku berhenti dan melihat-lihat sepanjang *fjord* – matahari terbenam, dan awan berubah menjadi semerah darah. Aku merasakan sebuah teriakan melalui alam; tampaknya aku mendengar teriakan itu. Aku melukis gambaran ini, melukis awan seperti darah yang sebenarnya. Warnanya berteriak. Hal ini menjadi *The Scream*, seperti pada Gambar 2.15.

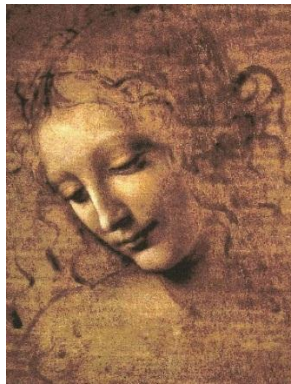


Gambar 2.15 Lukisan *The Scream*

2.12.3 *La Scapigliata*

La Scapigliata – juga dikenal sebagai *Head of a Woman* merupakan sebuah lukisan minyak yang dilukis pada kayu oleh master Renaisans Leonardo da Vinci, yang diselesaikan pada tahun 1508. Karya tersebut pertama kali diperkenalkan di koleksi Rumah Gonzaga pada tahun 1627. Lukisan ini dibuat pada masa Renaisans Tertinggi. Diantara banyak karya seni dari Leonardo, *La Scapigliata* merupakan salah satu yang dapat dianggap tidak lazim pada masanya. Para kritikus berpendapat bahwa Leonardo tidak hanya membuat sebuah sketsa dari seorang wanita dengan rambut yang tidak tersisir, malahan mereka berpendapat bahwa dia telah membuat sebuah karya seni yang

melukiskan kecantikan alami dan kekuatan inheren dari wanita. *La Scapigliata* menjaga rasa kesetaraan antara pria dan wanita dimana kesetaraan sendiri belum ada. Melalui *La Scapigliata*, Leonardo berpendapat bahwa ada sesuatu yang lebih dari wanita dari yang dipahami dan dihargai oleh masyarakat. Dia menggunakan garis-garis halus dan bayangan yang lembut pada sketsanya yang mengilustrasikan bagaimana seorang wanita itu cantik secara alami. Konsep dari rambut yang terkesan liar memberi kesan kekuatan asli dari perempuan yang jarang diperlihatkan. Ilustrasi lukisan *La Scapigliata* dapat dilihat pada Gambar 2.16.



Gambar 2.16 Lukisan *La Scapigliata*

2.12.4 Patung *Liberty*

Patung *Liberty* merupakan sebuah patung raksasa bergaya neoklasikal yang terletak pada Pulau *Liberty* di Pelabuhan New York di Kota New York, Amerika Serikat. Patung berbahan perunggu tersebut merupakan sebuah hadiah dari rakyat Perancis kepada rakyat Amerika Serikat. Patung ini didesain oleh pematung Perancis Frédéric Auguste Bartholdi dan dibangun oleh Gustave Eiffel dan diresmikan pada 28 Oktober 1886. Patung *Liberty* merupakan sesosok wanita berjubah yang merepresentasikan *Libertas*, salah seorang dewi Romawi. Dia memegang obor pada tangan kanannya, dan di tangan kirinya dia membawa sebuah *tabula ansata* yang bertuliskan “July 4, 1776”, tanggal dari Deklarasi Kemerdekaan Amerika. Sebuah rantai rusak terbaring pada kakinya. Patung ini menjadi ikon dari kemerdekaan dan Amerika Serikat, dan merupakan sebuah simbol selamat datang bagi

imigran yang datang dari luar negeri. Bartholdi terinspirasi dari profesor hukum dan politikus dari Perancis Édouard René de Laboulaye, yang pernah berkomentar pada tahun 1865 bahwa sebuah monumen yang dihadiahkan untuk kemerdekaan Amerika akan menjadi proyek bersama yang tepat bagi rakyat Perancis dan Amerika. Gambar 2.17 merupakan foto dari Patung *Liberty*.



Gambar 2.17 Patung *Liberty*

2.12.5 Menara Eiffel

Menara Eiffel adalah sebuah menara jaring-jaring besi di Champ de Mars di Paris, Perancis. Menara ini dinamakan berdasarkan insinyur pembuatnya, Gustave Eiffel. Dibangun pada tahun 1887-1889 sebagai ikon *World Fair* 1889, menara ini pada awalnya dikritik oleh beberapa seniman terkemuka dan cendekiawan dari Perancis karena desainnya, namun menara ini telah menjadi ikon budaya global dari Perancis dan salah satu dari struktur yang paling terkenal di dunia. Menara Eiffel adalah monumen berbayar yang paling sering dikunjungi di dunia; 6,91 juta orang telah menaikinya pada tahun 2015. Menara ini memiliki tinggi 324 meter, kira-kira setinggi gedung bertingkat 81 lantai, dan merupakan struktur tertinggi di Paris. Dasarnya berbentuk kotak, dengan ukuran 125 meter pada setiap sisinya. Menara ini memiliki 3 tingkat untuk pengunjung, dengan restoran pada tingkat pertama dan kedua. Balkon atas yang terletak pada tingkat teratas

berjarak 276 meter dari tanah – dek observasi tertinggi yang dapat diakses publik di Uni Eropa, yang dapat dilihat dari foto menara Eiffel pada Gambar 2.18.



Gambar 2.18 Menara Eiffel

----Halaman ini sengaja dikosongkan----

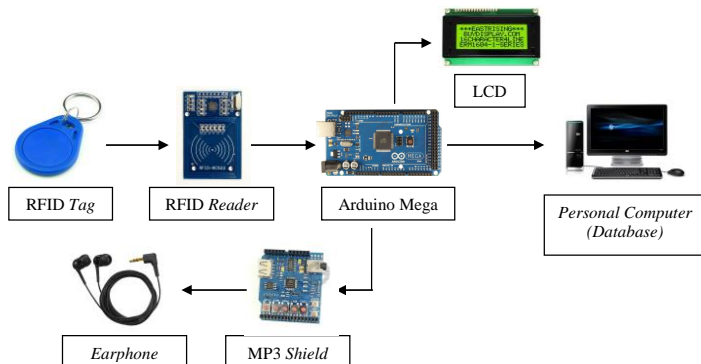
BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini membahas perancangan sistem dan alat yang akan dibuat. Dimana akan dibagi menjadi dua bagian, untuk Cahyo Dhian Tyastono bertugas untuk merancang dan membuat antarmuka pada alat yang berhubungan dengan LCD serta menyimpan dan membaca data nama obyek pada EEPROM yang datanya ditampilkan pada antarmuka pada PC dengan menggunakan *software Visual Basic*. Untuk Dany Prasetyo bertugas untuk merancang dan membuat *hardware* rangkaian *keypad* yang berfungsi untuk tombol memilih bahasa , menambah dan mengurangi volume MP3 dan untuk menghentikan MP3, membuat *software* identifikasi obyek dengan menggunakan teknologi RFID yang terdiri dari modul *reader* MFRC522 dan *tag* RFID dan membuat sistem penyimpanan data narasi obyek pada *SD Card Mp3 Shield*.

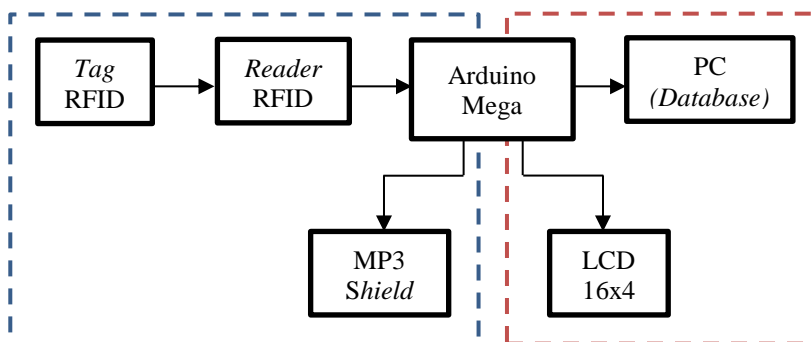
3.1 Gambaran Umum Sistem

Spesifikasi sistem yang diharapkan tercapai pada perancangan Tugas Akhir ini adalah mengidentifikasi obyek yang terdapat pada museum dalam bentuk narasi audio dengan format MP3 dan membuat sistem *database* nama obyek yang telah teridentifikasi oleh RFID dengan menggunakan basis data *Visual Basic*. Arsitektur sistem yang dirancang ditunjukkan pada Gambar 3.1



Gambar 3.1 Arsitektur Sistem

Dari arsitektur sistem yang dirancang bagian untuk sistem identifikasi obyek museum menggunakan teknologi RFID dan modul MP3 *shield* yang dikendalikan oleh *Arduino* yang dikerjakan oleh Dany Prasetyo. Untuk sistem *database* nama obyek museum pada *Arduino* dan dikomunikasikan dengan PC untuk ditampilkan pada *software Visual Basic* serta tampilan LCD pada alat yang dikerjakan oleh Cahyo Dhian Tyastono. Untuk diagram fungsional dari sistem identifikasi obyek keseluruhan dapat dilihat pada Gambar 3.2



Keterangan :

--- : Cahyo Dhian Tyastono

--- : Dany Prasetyo

Gambar 3.2 Diagram Fungsional Keseluruhan Sistem

Untuk mendapatkan alat pemutar narasi tentang obyek di museum yang berjalan sesuai harapan diperlukan bagian pokok, yaitu:

1. Komponen RFID, yang terdiri dari *reader* dan *tag* berfungsi sebagai alat untuk melakukan proses identifikasi dengan gelombang radio pada obyek yang akan digunakan.
2. *Arduino Mega*, mikrokontroler yang berfungsi sebagai pengendali utama dalam sistem ini dimana *Arduino* akan mengolah seluruh proses pengolahan data yang diterima dari *reader* dan MP3 *shield*. Pin yang digunakan adalah pin *digital input*.
3. LCD 16x4, yang berfungsi sebagai tempat untuk menampilkan hasil proses pengolahan data dari RFID ke *Arduino* kemudian

akan ditampilkan ke LCD dan dilengkapi dengan beberapa fitur tambahan jam dan menu pilihan.

4. Modul MP3 *Shield*, berfungsi untuk menyimpan *file* dalam bentuk MP3 dan mengeluarkan *file* audio tersebut dalam bentuk suara sesuai dengan perintah dari *Arduino*.
5. PC (*Database*), berfungsi untuk menyimpan jumlah hasil pembacaan dari RFID pada alat yang selanjutnya digunakan untuk melihat berapa kali obyek museum dilihat oleh pengunjung pada komputer.

Adapun penjelasan dari diagram fungsional pada Gambar 3.2 dan juga tahapan untuk pemutaran narasi obyek sehingga mencapai penjelasan tentang obyek yang telah teridentifikasi. Penjelasan dari masing-masing bagian dijelaskan pada sub bab perancangan perangkat elektronik, perancangan perangkat keras, dan perancangan perangkat lunak.

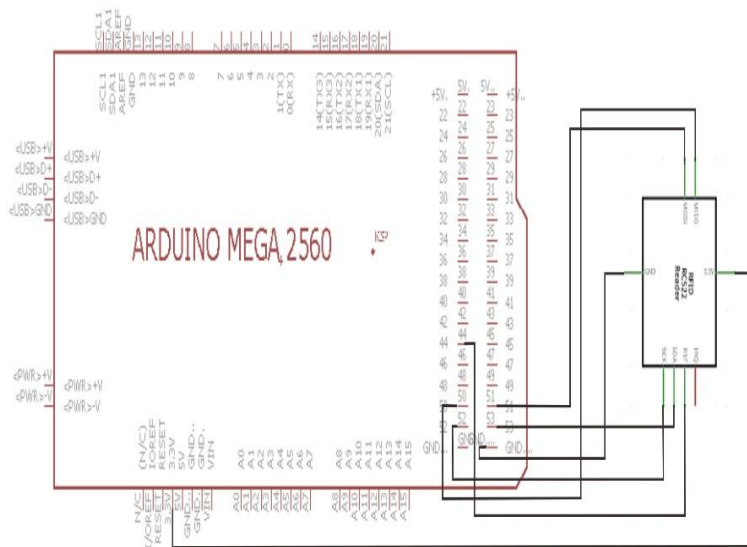
3.2 Perancangan Perangkat Keras

Tahap pertama dalam perancangan sistem adalah perancangan perangkat keras yang meliputi pembuatan *design box*, pembuatan *display* obyek museum, pengkabelan *Arduino Mega 2560* dengan *reader* MFRC522, rangkaian RTC, pengkabelan rangkaian RTC dengan *Arduino Mega 2560*, pengkabelan LCD dan I2C dengan *Arduino Mega*, rangkaian *push button*, pengkabelan *push button* dengan *Arduino Mega*, dan pengkabelan MP3 *Shield* dengan *Arduino Mega 2560*. Perangkat elektronik di dalamnya ini terdiri dari beberapa komponen yang dirancang dan disesuaikan berdasarkan kebutuhan yang diperlukan pada alat pemutar narasi audio ini. Komponen yang diperlukan pada alat pemutar narasi audio ini diantaranya adalah modul mikrokontroler, MP3 *player*, RFID *reader*, LCD, modul I2C, RTC, dan *push button*. Modul mikrokontroler yang digunakan adalah *Arduino Mega 2560*, MP3 *player* yang digunakan adalah MP3 *Shield Arduino*, RFID *reader* yang digunakan adalah MFRC522, LCD dengan karakter 16x4, modul I2C untuk LCD dengan IC PCF8574, dan RTC dengan memakai IC DS1307.

3.2.1 Pengkabelan Arduino Mega 2560 dengan Reader MFRC522

Perancangan pengkabelan untuk *reader* MFRC522 dengan *Arduino Mega 2560* merupakan fokus pekerjaan dari Dany Prasetyo. *Reader* RFID yang digunakan adalah modul *reader* jenis MFRC522.

Modul *reader* MFRC522 ini digunakan sebagai proses *scanning* atau pembacaan UID pada *tag* dengan benar dan mengkomunikasikan hasilnya ke mikrokontroler. Pada pembuatan alat ini modul *reader* MFRC522 untuk membaca dan menulis data dari *tag* ke *Arduino* menggunakan komunikasi serial SPI dimana pin-pin yang terdapat pada modul *reader* MFRC522 dihubungkan dengan pin *digital input* *Arduino*. Untuk lebih jelasnya pengkabelan *reader* MFRC522 dengan *Arduino Mega 2560* dapat dilihat pada Gambar 3.3.



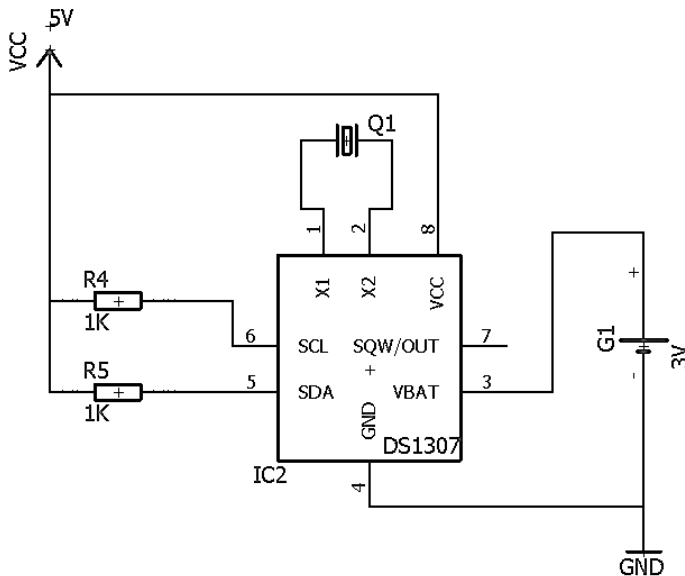
Gambar 3.3 Skema Pengkabelan *Arduino Mega 2560* dengan *Reader MFRC522*

Pin-pin yang digunakan pada *Arduino Mega 2560* dengan *Reader MFRC522* pada sistem ini, yaitu:

- Pin D50 => MISO
- Pin D51 => MOSI
- Pin D52 => SCK
- Pin D53 => SDA/NSS
- Pin D44 => *Reset*
- 3,3V => Sumber tegangan untuk *Reader MFRC522*
- GND

3.2.2 Rangkaian RTC

Perancangan untuk membuat rangkaian pewaktu menggunakan RTC merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Rangkaian RTC ini digunakan sebagai penyedia waktu dengan menggunakan IC DS1307. Pada rangkaian ini selain IC DS1307 juga dilengkapi dengan resistor 2 buah $1k\Omega$, baterai 3V untuk suplai tegangan sekunder serta *header* untuk tempat kabel yang akan menyambungkan rangkaian dengan *Arduino*. Rangkaian RTC dapat dilihat pada Gambar 3.4.

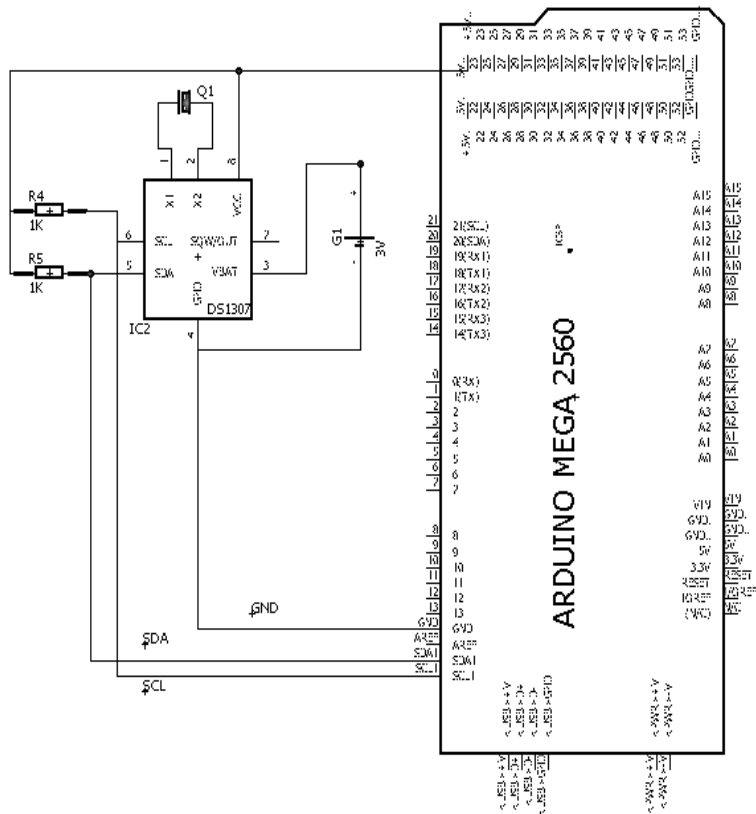


Gambar 3.4 Rangkaian RTC DS1307

3.2.3 Pengkabelan Rangkaian RTC dengan *Arduino Mega 2560*

Pengkabelan rangkaian RTC dengan *Arduino Mega 2560* merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Rangkaian RTC DS1307 disambungkan dengan *Arduino Mega* dengan menggunakan *header*. Untuk suplai tegangan 5V RTC, pin VCC dihubungkan dengan pin 5V. Pin GND pada RTC dihubungkan dengan pin GND pada *Arduino*. Sebagai jalur data dan *clock*, pin SDA dan SCL pada RTC dihubungkan dengan pin SDA dan SCL pada *Arduino*.

Pengkabelan dari rangkaian RTC dengan *Arduino Mega 2560* dapat dilihat pada Gambar 3.5.

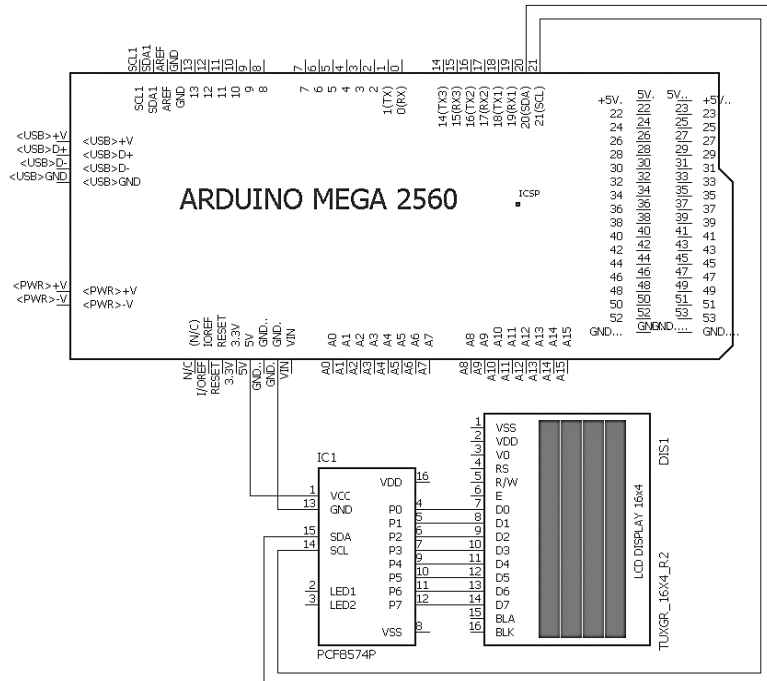


Gambar 3.5 Skema Pengkabelan RTC dengan *Arduino Mega 2560*

3.2.4 Pengkabelan LCD dan I2C dengan *Arduino Mega 2560*

Perancangan pengkabelan LCD 16x4 dan I2C dengan *Arduino Mega 2560* merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. LCD dihubungkan dengan *Arduino Mega* dengan menggunakan komunikasi data serial yang dilakukan melalui I2C (*Inter Integrated Circuit*). Semua pin yang ada pada LCD dihubungkan dengan pin pada I2C untuk LCD secara permanen menggunakan solder. Pin SCL dan SDA dari I2C disambungkan pada pin SCL dan SDA pada *Arduino*

Mega 2560 sebagai jalur data dan jalur *clock*. Pin VCC dan GND pada I2C dihubungkan dengan pin 5V dan GND pada *Arduino Mega 2560* sebagai *input* sumber daya untuk LCD. Pengkabelan dari LCD dan I2C dengan *Arduino Mega* untuk lebih jelasnya dapat dilihat pada Gambar 3.6.

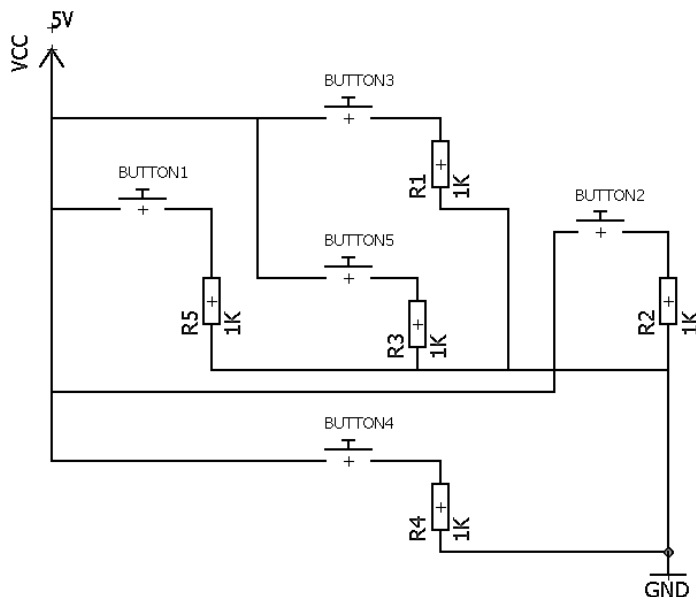


Gambar 3.6 Pengkabelan LCD, I2C dengan *Arduino Mega 2560*

3.2.5 Rangkaian Keypad

Perancangan rangkaian *keypad* merupakan fokus pekerjaan dari Dany Prasetyo. Rangkaian *keypad* sebagai saklar ini berfungsi sebagai tombol menu untuk memilih bahasa dan menrubah nilai *volume* pada *MP3 shield*. Rangkaian ini terdiri dari resistor 1k Ω dan *push button*. Jika rangkaian ini diberi logika 1 atau ditekan salah satu tombol *push button*, maka rangkaian ini akan berkerja sesuai fungsi masing-masing *push button* yang digunakan. Rangkaian ini juga terhubung dengan sumber

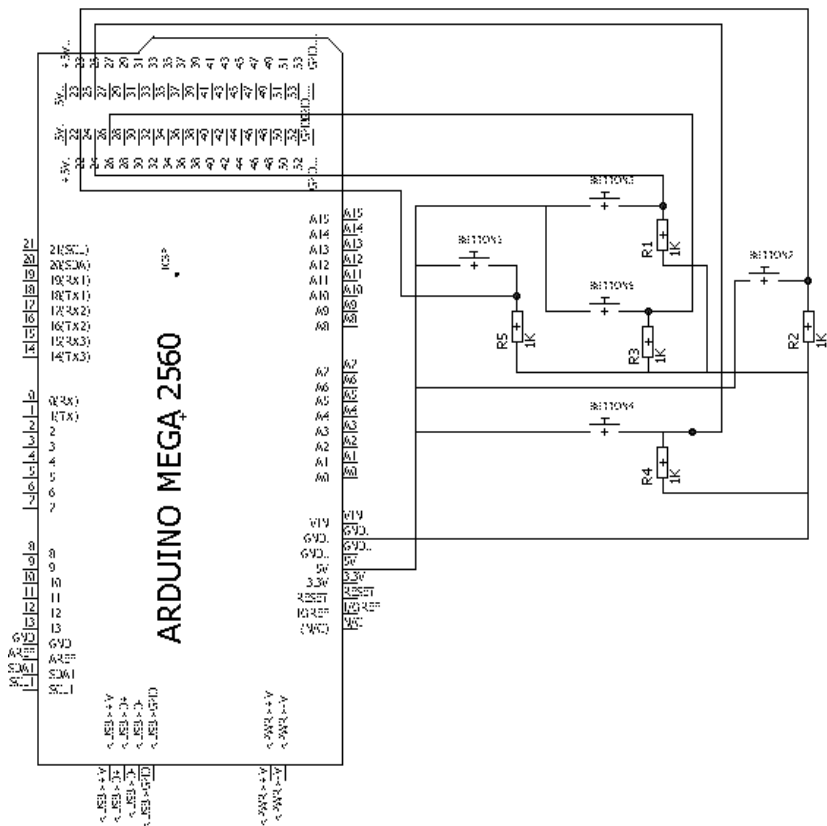
tegangan 5V. Rangkaian *push button* yang digunakan dapat ditunjukkan pada Gambar 3.7.



Gambar 3.7 Skema Rangkaian Keypad

3.2.6 Pengkabelan Keypad dengan Arduino Mega 2560

Perancangan pengkabelan rangkaian *keypad* dengan *Arduino Mega 2560* merupakan fokus pekerjaan dari Dany Prasetyo. *Push Button* yang digunakan adalah *push button* dengan 4 pin kaki. *Push button* pada sistem identifikasi obyek ini berfungsi sebagai tombol tekan untuk memilih beberapa kondisi yang diinginkan oleh pengguna alat ini. Pada sistem ini terdapat 5 buah *push button* dengan fungsi yang berbeda-beda yang telah dihubungkan dengan pin *digital input* *Arduino* untuk dapat memberikan kondisi ketika button ditekan maka *Arduino* akan menerima data dari *push button* dan mengolah data tersebut. Untuk lebih jelasnya Pengkabelan *push button* dengan *Arduino Mega 2560* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Pengkabelan Keypad dengan Arduino Mega 2560

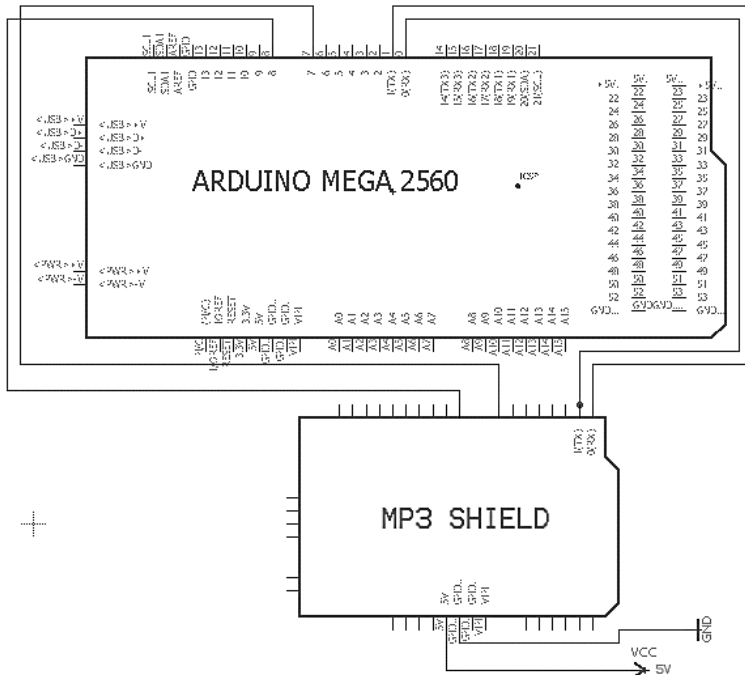
Pin-pin yang digunakan pada *Arduino Mega 2560* pada sistem ini, yaitu:

- Pin D22 => *Button 1* (Bahasa Indonesia)
- Pin D23 => *Button 2* (Bahasa Inggris)
- Pin D24 => *Button 3* (Volume +)
- Pin D25 => *Button 4* (Volume -)
- Pin D26 => *Button 5* (Stop)
- 5V => Sumber tegangan untuk *Button*
- GND

3.2.7 Pengkabelan MP3 Shield dengan Arduino Mega 2560

Perancangan pengkabelan MP3 *shield* dengan *Arduino Mega 2560* merupakan fokus pekerjaan dari Dany Prasetyo. Pada perancangan perangkat elektronika sistem terdapat mikrokontroler *Arduino Mega 2560* yang terhubung dengan modul MP3 *shield*, *Arduino Mega 2560* berfungsi sebagai kontroler dari sistem, sedangkan MP3 *shield* berfungsi untuk memecahkan kode *file* audio dan untuk menerima *file input* melalui bus *input* serial (SPI). Setelah *file* itu diterjemahkan, kemudian audio tersebut dikirim keluar ke *stereo headphone* yang menggunakan *jack* 3,5mm, serta 2-pin 0,1 " *pitch header*. Mp3 *shield* arduino ini memiliki banyak kelebihan seperti kontrol oleh antarmuka serial UART atau tombol *onboard*, *have build in* Audio amplifier 3W, dan memiliki *On board* 64MBit SPI-FLASH, yang dapat simpan *file audio* pada SD-card yang terdapat *board* MP3, dan dimainkan nanti.

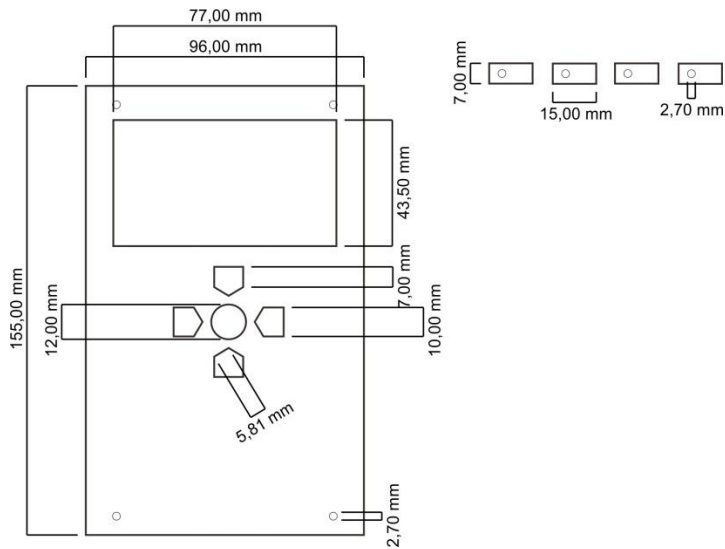
Pin-pin yang digunakan oleh modul MP3 *shield* telah dibuat dengan menyesuaikan dengan mikrokontroler *Arduino Mega 2560*. Untuk pemasangan dari modul MP3 *shield* ke *Arduino Mega 2560* hanya ditancapkan sejak mulai dari pin 0 (RX0) hingga AREF dan pin A5 hingga *RESET*, *shield* ini bisa dikontrol dari arduino dengan serial *hardware* Rx D0, Tx D1 dan dengan *software* Serial RX D7, Tx D8. Berikut gambar dari pemasangan arduino mega 2560 dan modul MP3 *shield* pada Gambar 3.9.



Gambar 3.9 Skematik Modul MP3 Shield dengan Arduino Mega 2560

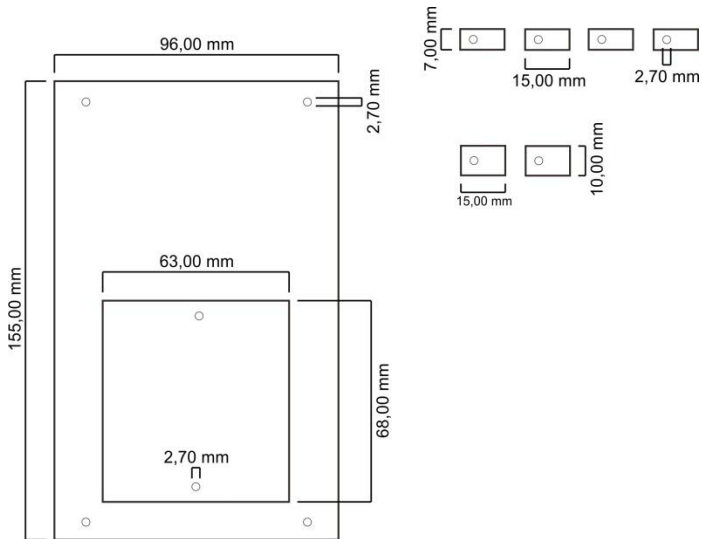
3.2.8 Perancangan *Design Box*

Perancangan untuk pembuatan *box* merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. *Box* ini digunakan sebagai tempat dari rangkaian Arduino dengan MP3 shield, reader RFID, RTC, tombol-tombol, LCD dan komponen-komponen pendukung yang lainnya yang merupakan rangkaian inti dari alat yang dibuat. Bahan yang digunakan untuk membuat *box* ini terbuat dari bahan akrilik dengan ketebalan 3mm. *Box* ini disambungkan dengan menggunakan lem untuk akrilik dan juga menggunakan mur dan baut.



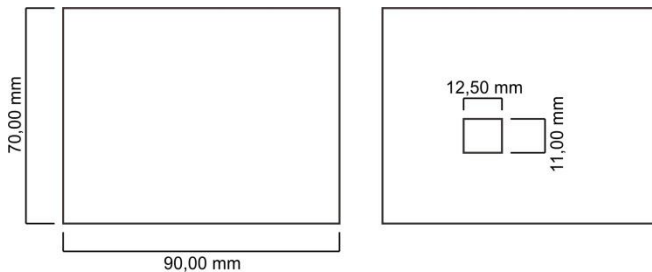
Gambar 3.10 Perancangan *Box* Tampak Depan

Rangka untuk *box* bagian depan yang ditunjukkan pada Gambar 3.10 ini digunakan sebagai tempat dari LCD untuk menampilkan nama obyek, jam dan antarmuka dari menu yang dapat diakses dari tombol-tombol yang terletak di bagian bawah LCD yang memiliki fungsi untuk mengatur *volume* suara narasi, memilih bahasa untuk narasi yang dimainkan serta untuk menghentikan pemutaran narasi. Pada bagian ujung-ujungnya terdapat empat lubang untuk menyambungkan bagian depan dengan rangka bagian atas dan samping dengan baut dan mur melalui empat persegi kecil yang ditempelkan pada rangka bagian samping dengan lubang pada masing-masing persegi yang sejajar dengan lubang pada ujung-ujung rangka depan.



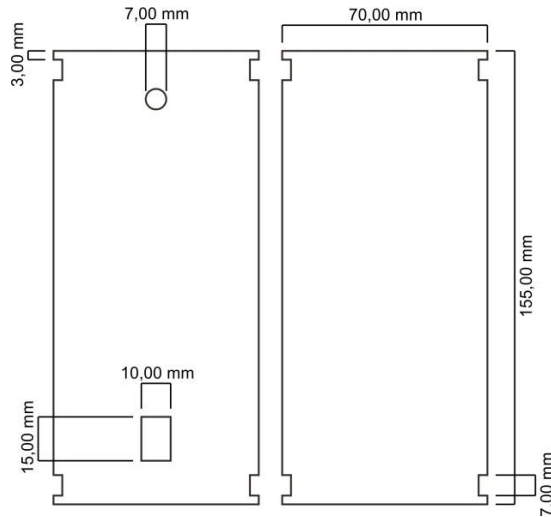
Gambar 3.11 Perancangan *Box* Tampak Belakang

Rangka *box* bagian belakang yang ditunjukkan pada Gambar 3.11 memiliki dimensi yang sama dengan rangka bagian depan tetapi untuk bagian belakang ini digunakan untuk tempat dari penutup baterai. Untuk dapat disambungkan dengan bagian samping cara yang digunakan sama dengan rangka bagian depan.



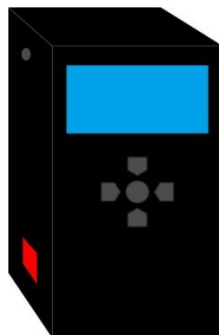
Gambar 3.12 Perancangan *Box* Tampak Atas dan Tampak Bawah

Rangka bagian atas dan bawah yang ditunjukkan pada Gambar 3.12 memiliki dimensi dan bentuk persegi panjang yang serupa. Tetapi untuk rangka bagian bawah memiliki lubang untuk tempat koneksi USB antara *Arduino* dengan dengan PC, sedangkan untuk rangka bagian atas hanya digunakan sebagai penutup bagian atas.



Gambar 3.13 Perancangan *Box* Tampak Samping

Rangka bagian samping seperti pada Gambar 3.13 ini berbentuk persegi panjang dengan cekungan di ujung-ujungnya. Cekungan ini berfungsi untuk menempelkan persegi panjang kecil untuk menyambungkan bagian samping dengan rangka bagian depan dan belakang dengan baut dan mur. Rangka yang diletakkan pada sebelah kanan hanya berfungsi sebagai penutup dan untuk bagian kiri terdapat lubang berbentuk lingkaran untuk tempat *jack* audio dan lubang berbentuk persegi panjang untuk tempat *switch power*. *Box* yang sudah disusun dapat dilihat pada Gambar 3.14.

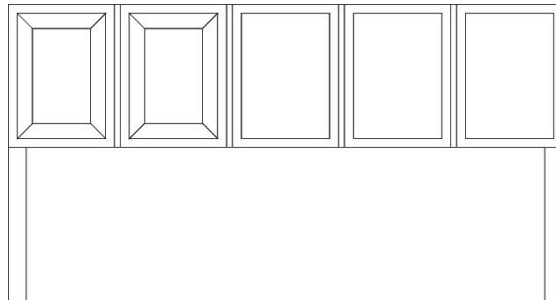


Gambar 3.14 Perancangan *Box* Keseluruhan Ketika Disatukan

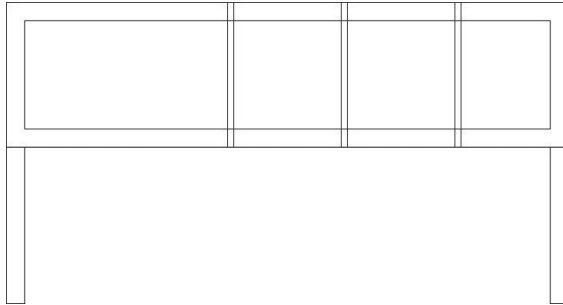
3.2.9 Perancangan *Display* Obyek Museum

Perancangan untuk pembuatan *display* obyek museum merupakan fokus pekerjaan dari Dany Prasetyo. Untuk menampilkan obyek museum, diperlukan suatu tempat untuk menempatkan obyek-obyek museum agar dapat ditampilkan. Sebagai tempat untuk obyek museum, dibuatlah *display* yang dapat menampung 5 buah obyek museum, yang terdiri dari 3 buah lukisan yang kita gunakan berupa poster dan patung 2 buah.

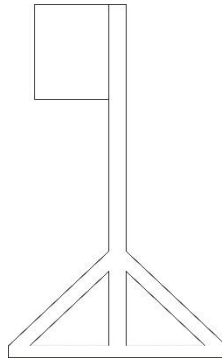
Perancangan untuk *display* obyek museum merupakan fokus pekerjaan dari Dany Prasetyo. *Display* untuk tempat obyek museum ini rangkanya dibuat dari kayu dengan tinggi 1 m dan lebar 1,8 m. Tempat untuk masing-masing obyek dibuat untuk kertas ukuran A3, dan untuk obyek patung diberi *box* untuk ruangan *display*nya yang memiliki panjang ke belakang sepanjang 25 cm. Perancangan dari *display* ini dapat dilihat pada Gambar 3.15 untuk tampak depannya, untuk tampak belakang ditunjukkan pada Gambar 3.16, dan Gambar 3.17 untuk tampak samping.



Gambar 3.15 Perancangan *Display* Obyek Tampak Depan



Gambar 3.16 Perancangan *Display* Obyek Tampak Belakang



Gambar 3.17 Perancangan *Display* Obyek Tampak Samping

3.3 Perancangan Perangkat Lunak

Setelah melakukan perancangan perangkat keras, selanjutnya diperlukan sebuah perancangan perangkat lunak yang mampu mengelola, mengontrol peralatan pendukung lainnya dan berkomunikasi dengan *software* lainnya untuk mengirim dan menerima data. *Software* program berisi perintah – perintah yang dieksekusi oleh modul *arduino* sehingga sistem dapat bekerja sesuai dengan alur dan tujuan yang direncanakan. Perancangan perangkat lunak (*software*) yang digunakan yaitu dengan menggunakan program *Arduino IDE (Integrated Development Environment)*, program yang dibuat meliputi program scanning *tag* RFID menggunakan modul *reader* MFRC522, program untuk pemilihan Bahasa, program untuk menghentikan MP3 ketika sedang berbunyi.

Flowchart merupakan bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Bagan ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas.

Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- *START*: berisi instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- *READ*: berisi instruksi untuk membaca data dari suatu peralatan.
- *PROCESS*: berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- *WRITE*: berisi instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- *END*: mengakhiri kegiatan pengolahan

3.3.1 Flowchart Sistem Scanning Tag RFID

Perancangan sistem *scanning tag* RFID merupakan fokus pekerjaan dari Dany Prasetyo. Sistem *scanning tag* RFID dapat digambarkan dengan sebuah *flowchart* pada Gambar 3.18. Penjelasan dari *flowchart* sistem *scanning reader* terhadap *tag* RFID sebagai berikut:

1. *Reader* MFRC522 akan menginialisasi *tag* yang berada dalam jangkauan baca *reader*.
2. Kemudian *tag* akan memberikan sinyal dan mengirimkan UID *tag* pada *reader* dalam bentuk *string*.
3. *Buzzer* akan berbunyi sebagai tanda bahwa proses *scanning reader* terhadap *tag* sudah berhasil didapatkan nilai UID.
4. Pada modul MP3 *shield* terdapat sebuah SD-card yang digunakan untuk menyimpan file mp3, ketika UID *tag* telah didapatkan kemudian UID *tag* akan disesuaikan sesuai alamat yang telah tersimpan pada SD-card jika UID *tag* tidak terdaftar pada SD-card maka *reader* akan kembali mengidentifikasi *tag* RFID

3.3.2 Flowchart Perancangan Sistem Pemilihan Bahasa dan Menyimpan Nama Obyek ke EEPROM Arduino

Perancangan sistem perancangan pemilihan bahasa merupakan fokus pekerjaan dari Dany Prasetyo. Secara umum alat ini berupa *keypad* yang terdiri dari beberapa *push button* yang dirangkai yang dihubungkan dengan pin *digital input* pada *Arduino Mega*. Sistem perancangan sistem pemilihan bahasa dapat digambarkan dengan sebuah *flowchart* di Gambar 3.19. Penjelasan dari *flowchart* sistem *scanning reader* terhadap *tag* RFID sebagai berikut:

1. Pada layar LCD 16x4 menampilkan menu pilihan Bahasa yaitu Bahasa Indonesia dan Bahasa Inggris.
2. Pada *keypad* terdapat 2 tombol yaitu tombol A untuk memilih pilihan Bahasa Indonesia dan tombol B untuk memilih Bahasa Inggris.
3. Ketika telah terpilih bahasa yang diinginkan maka file mp3 dengan daftar narasi yang berbahasa Indonesia akan dimainkan.
4. Kemudian nama obyek yang telah teridentifikasi akan tersimpan di EEPROM.

3.3.3 Flowchart Perancangan Sistem untuk Menghentikan MP3 Shield

Perancangan sistem pemilihan untuk menghentikan file MP3 merupakan fokus pekerjaan dari Dany Prasetyo. Alat ini menggunakan *keypad* yang terdiri dari beberapa button, untuk menghentikan file MP3 ketika dimainkan yaitu dengan menekan tombol stop. Sistem perancangan sistem untuk menghentikan MP3 *shield* dapat digambarkan dengan sebuah *flowchart* pada Gambar 3.20. Penjelasan dari sistem perancangan sistem untuk menghentikan MP3 *shield* sebagai berikut:

1. Setelah proses pemilihan bahasa dilakukan maka *file* MP3 yang dimainkan hanya file dengan format bahasa yang terpilih.
2. Ketika *file* MP3 sedang dimainkan untuk menghentikan suara narasi, tekan tombol *stop* untuk menghentikan suara narasi.
3. Kemudian *file* MP3 akan berhenti.

3.3.4 *Flowchart* Prosedur *Set-Up* Komunikasi Serial *Arduino* dengan PC

Perancangan sistem untuk melakukan komunikasi serial antara *Arduino* dengan PC dengan *platform Visual Basic* merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Prosedur *set-up* komunikasi serial *Arduino* dengan PC atau komputer dapat digambarkan dengan sebuah *flowchart* di Gambar 3.21. Penjelasan dari *flowchart* prosedur *set-up* komunikasi serial adalah sebagai berikut:

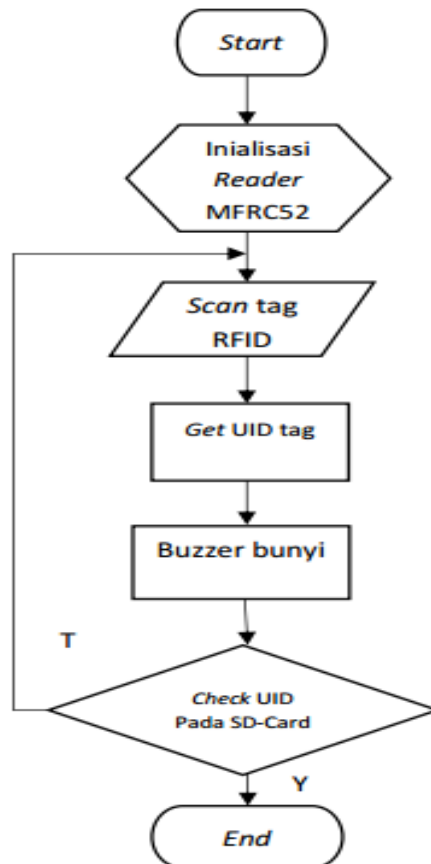
1. Setelah alat disambungkan dengan komputer, pada antarmuka di *Visual Basic* pilih COM untuk *port* yang digunakan untuk komunikasi serial serta nilai *baud rate* yang digunakan.
2. Kemudian tekan tombol *CONNECT* pada antarmuka untuk menyambungkan komunikasi serial.
3. Jika COM yang digunakan salah, maka akan muncul *message box* yang berisi pesan “Port Salah!” dan ketika COM yang digunakan sudah benar serta nilai *baud rate* yang dipilih sudah cocok dengan *baud rate* pada *Arduino* maka komunikasi serial sudah tersambung.

3.3.5 *Flowchart* untuk Menampilkan Data Nama Obyek pada Antarmuka *Visual Basic*

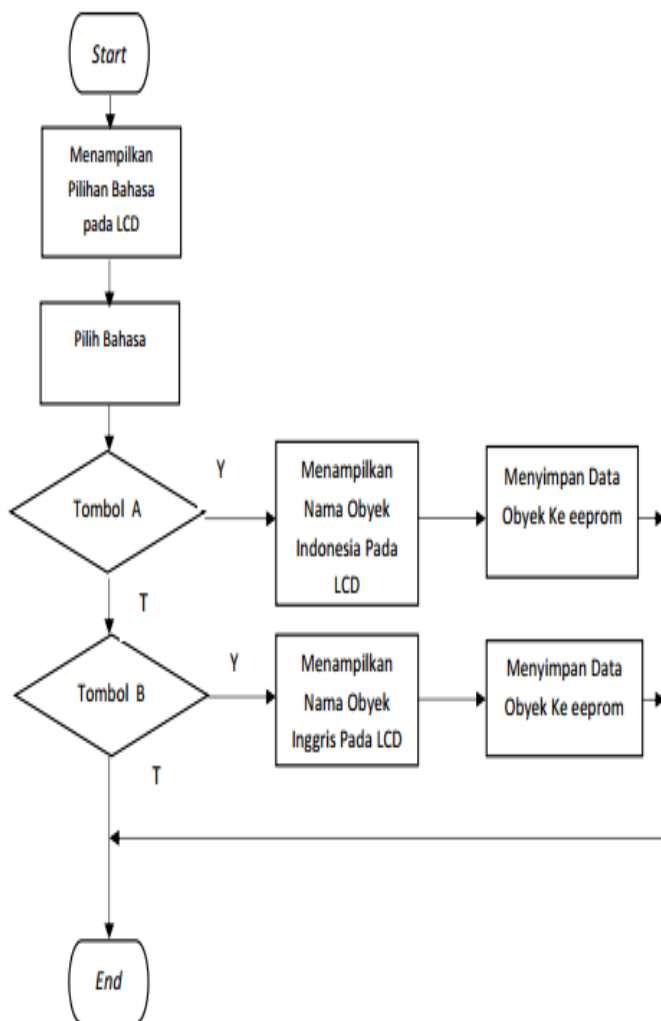
Perancangan sistem untuk menampilkan nama obyek sesuai data dari *Arduino* pada antarmuka *Visual Basic* merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Sistem untuk menampilkan nama obyek pada *Visual Basic* atau VB dapat digambarkan dengan sebuah *flowchart* pada Gambar 3.22. Penjelasan dari *flowchart* untuk menampilkan data nama obyek pada antarmuka VB adalah sebagai berikut:

1. Setelah data kode obyek dikirimkan dari EEPROM *Arduino* dan masuk ke PC, maka VB akan menerjemahkan kode obyek yang diterima agar lebih sederhana.
2. Data kode obyek yang telah disederhanakan akan dikirim ke *TextBox* *tbkode* dan *tbject* untuk dibandingkan dengan nilai yang terdapat pada *TextBox*.
3. Jika nilai kode yang masuk pada *TextBox* sama dengan nilai yang ada pada *TextBox* maka data tidak akan diproses. Jika nilainya berbeda maka data akan diproses dan kode obyek akan tersimpan di *tbkode* dan nama obyek akan tampil sesaat di *tbject*.

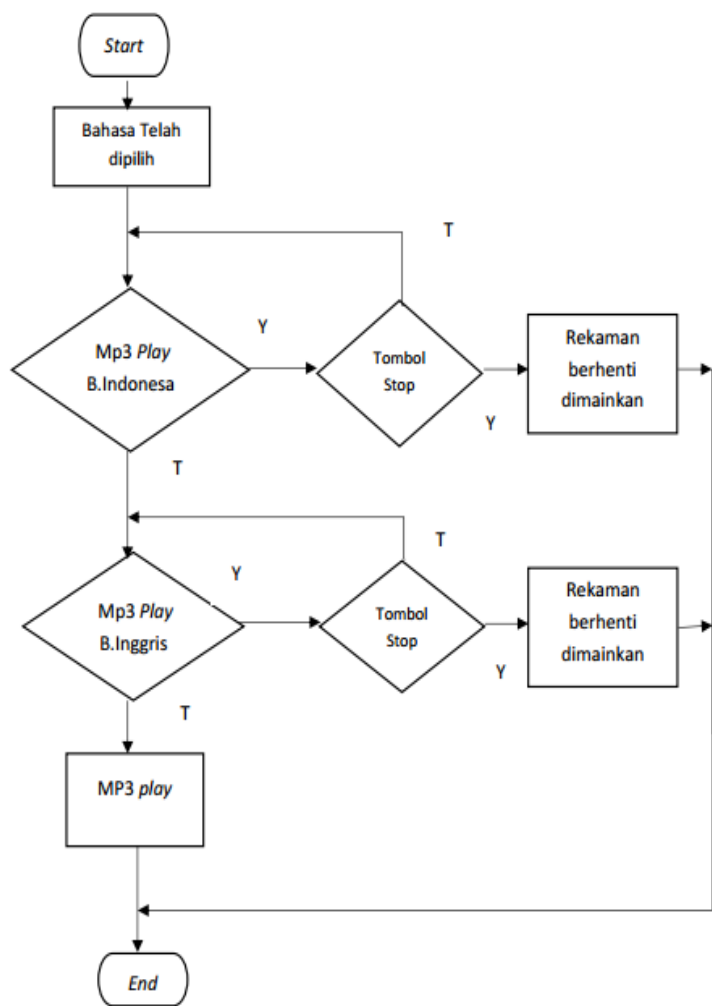
4. Kemudian data akan dikirim ke *Datagrid* dan nama obyek akan ditampilkan sesuai kode dari data yang masuk pada tabel Nama Obyek beserta jam dan tanggal pada tabel Jam dan Tanggal.
5. Ketika ada data selanjutnya maka data akan dimasukkan kembali ke *TextBox* untuk diproses dan proses akan berlanjut sampai semua data ditampilkan nama obyeknya.



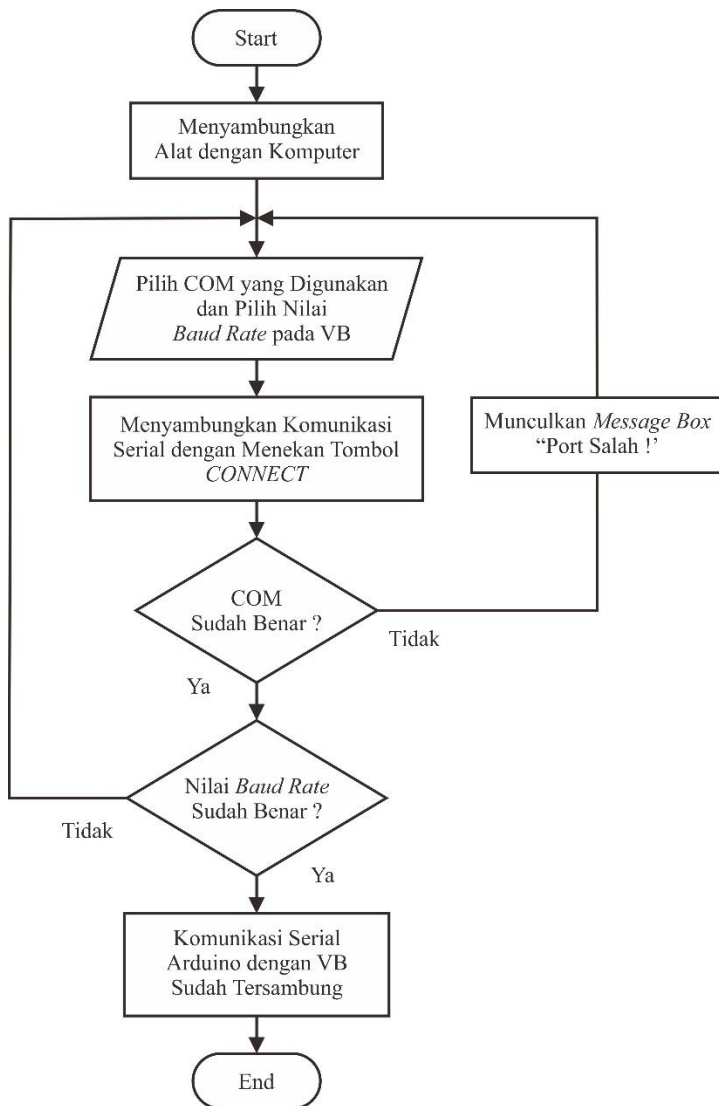
Gambar 3.18 Flowchart Sistem Scanning Tag RFID



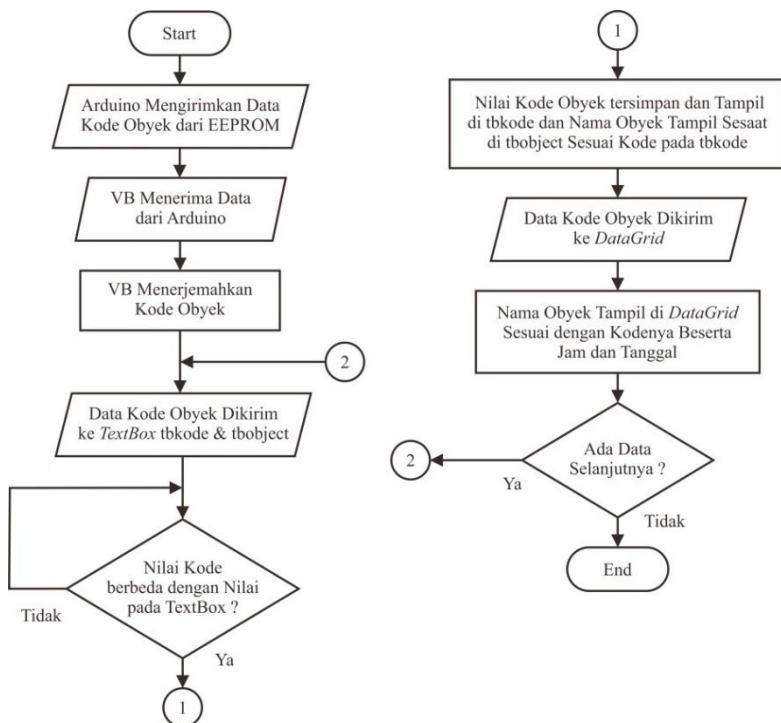
Gambar 3.19 Flowchart Sistem Pemilihan Bahasa



Gambar 3.20 Flowchart Sistem Menghentikan MP3 Shield



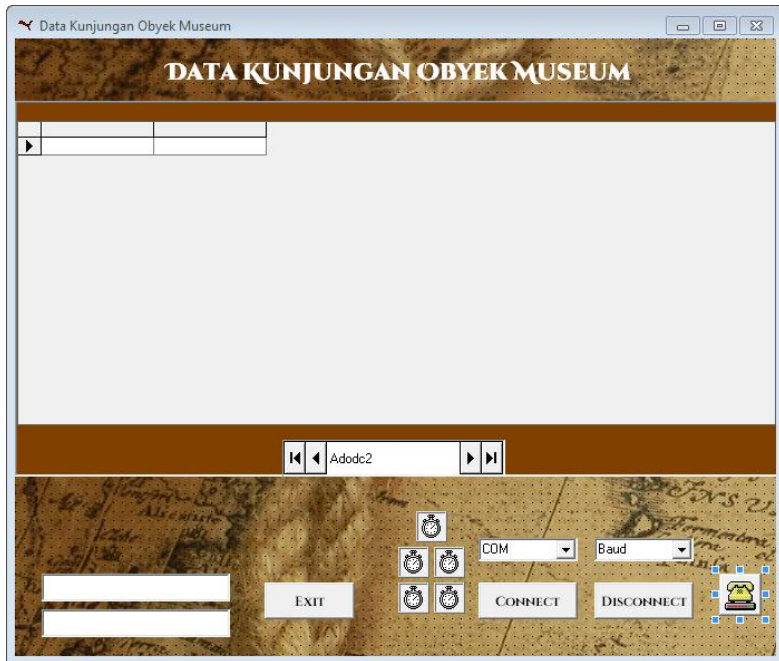
Gambar 3.21 Flowchart Sistem Komunikasi Serial Antara Arduino dengan PC dengan platform VB



Gambar 3.22 Flowchart Menampilkan Data Obyek Pada Antarmuka VB

3.3.6 Pembuatan Antarmuka Menggunakan Visual Basic 6

Pembuatan antarmuka menggunakan visual basic 6 merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Antarmuka ini merupakan aplikasi untuk menampilkan data dari mikrokontroler berupa nama obyek yang sudah disimak oleh pengunjung museum menggunakan *Visual Basic* yang nantinya akan disebut VB versi 6 seperti yang terlihat pada Gambar 3.23 tampilan antarmuka *visual basic* pada *form designer*. Kemudian data-data tersebut dapat disimpan di komputer. Perancangan antarmuka ini diawali dengan memasukkan komponen-komponen ke dalam *form designer*.



Gambar 3.23 Perancangan Antarmuka pada *Form Designer*

Pada *Form Designer* terdapat lima tombol yang terdiri atas, *CONNECT*, *DISCONNECT*, pemilihan *COM* dan *baud rate*, serta *EXIT*. *Datagrid* digunakan untuk menampilkan data yang terdiri atas baris dan kolom. Terdapat delapan komponen yang diambil dari *Toolbox* untuk diletakkan pada *form designer* seperti : *Datagrid*, *Adodc*, *Label*, *TextBox*, *CommandButton*, *ComboBox*, *Timer*, dan *MSComm*.

Datagrid digunakan untuk menampilkan semua data yang diterima dari Arduino dalam bentuk tabel yang diambil dari file *database* Microsoft Access. *Adodc* merupakan sumber data yang akan ditampilkan pada *Datagrid* yang terhubung dengan di Microsoft Access. *Label* digunakan untuk menambahkan teks pada *form*. *TextBox* berfungsi untuk menampilkan teks pada sebuah kotak. *CommandButton* merupakan suatu tombol yang berisi perintah. *ComboBox* adalah suatu komponen yang digunakan untuk memberikan pilihan. *Timer* digunakan sebagai pewaktu dan *MSComm* untuk komunikasi serial dengan Arduino.

---Halaman ini sengaja dikosongkan---

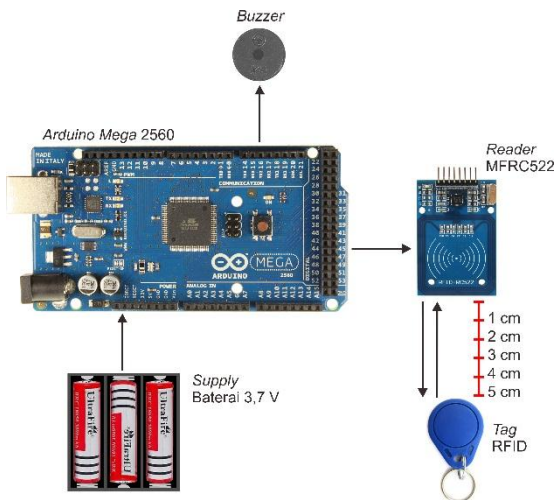
BAB IV

PENGUKURAN DAN PENGUJIAN

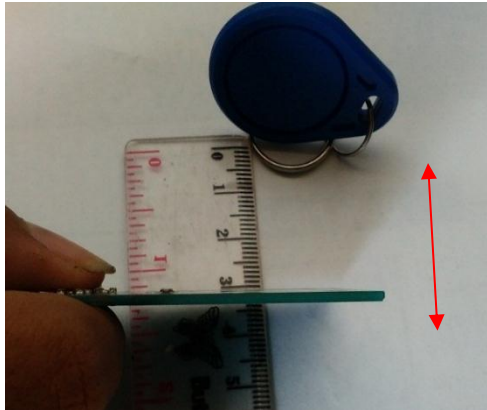
Untuk mengetahui bahwa alat telah bekerja dengan benar maka perlu dilakukan pengujian alat yang meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian yang dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat.

4.1 Pengukuran dan Pengujian RFID

Pengukuran dan pengujian sistem RFID merupakan fokus pekerjaan dari Dany Prasetyo. Tujuan dari pengukuran dan pengujian RFID adalah untuk mengetahui kerja *reader* dan *tag* RFID yang saling berkaitan dalam sistem kerja untuk pembuatan alat ini. Jenis *tag* yang digunakan pada Tugas Akhir ini adalah *Tag Key* dan *Reader* RFID yang digunakan adalah tipe MFRC522. Pemindaan UID *tag* dilakukan dengan menggunakan modul *reader* MFRC522 dengan skema untuk pengambilan data jarak baca *reader* pada *tag* seperti pada Gambar 4.1 berdasarkan program dengan alur seperti dalam *flowchart* pada Gambar 3.18.



Gambar 4.1 Skema Pengambilan Data jarak Baca Modul *Reader* pada *Tag*



Gambar 4.2 Pengujian Pengambilan UID Tag dalam jarak 1-3 cm

Cara pengujian alat untuk pengambilan data yang digunakan adalah seperti skema pada Gambar 4.1 menggunakan *supply* dari 3 buah baterai 3,7 V yang diseri sehingga tegangan *input* 11 V. Untuk mengukur jarak baca modul *reader* dilakukan pengukuran jarak dari jarak 1-5 cm dengan cara mendekatkan *tag* ke rangkaian *reader* seperti pada Gambar 4.2, apabila modul *reader* mendeteksi *tag* maka *buzzer* akan berbunyi dan itu menandakan bahwa *reader* berhasil mendeteksi UID *tag*. Sedangkan jika *buzzer* tidak berbunyi maka *reader* tidak berhasil mendeteksi *tag*, dengan hasil seperti pada Tabel 4.1.

Tabel 4.1 Hasil Data Jarak Pembacaan Tag

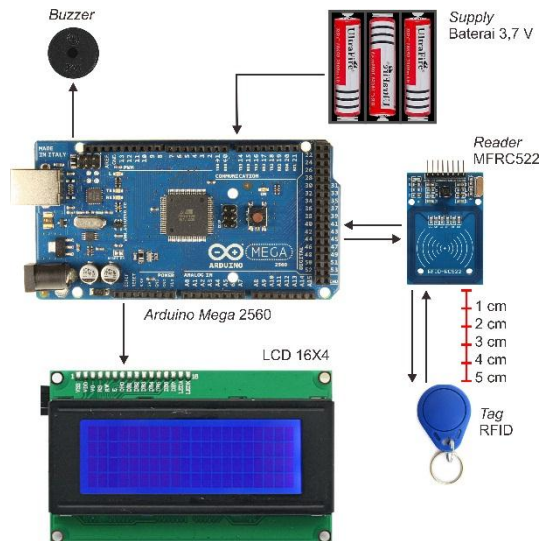
No	Jarak Tag dari Reader (cm)	Terdeteksi	Tak Terdeteksi
1	1 cm	✓	-
2	2 cm	✓	-
3	3 cm	✓	-
4	4 cm	-	✓
5	5 cm	-	✓

Dari hasil pengujian modul *reader* MFRC522 untuk mengetahui jarak baca *reader* agar dapat mengetahui UID *tag* dapat dilihat pada Tabel 4.1, diketahui bahwa *tag* dideteksi oleh *reader* pada jarak 1-3 cm

dan selanjutnya pada jarak 4-5 cm *reader* sudah tidak dapat mendeteksi *tag*. Dapat disimpulkan bahwa *reader* MFRC522 dapat berkerja dengan baik sesuai dengan spesifikasi yang diperlukan, yaitu jarak terdeteksinya *tag* oleh *reader* hanya sampai jarak 3 cm.

4.2 Pengujian RFID dengan LCD 16x4 untuk Menampilkan Nama Obyek

Pengujian RFID yang terdiri dari modul *reader* MFRC522 dan *tag* dengan LCD 16x4 dan *Arduino Mega* sebagai kendali utama pada sistem merupakan fokus pekerjaan Bersama. Tujuan dari pengujian RFID dengan LCD 16x4 dan *Arduino Mega* adalah untuk mengetahui apakah hubungan antara *Arduino Mega*, LCD 16x4, dengan modul *reader* MFRC522 dapat berkerja dengan baik atau tidak dan untuk mengetahui nomer UID *tag* yang dikirim oleh *reader* MFRC522 ke *Arduino*. Untuk mendukung pengujian ini diperlukan peralatan antara lain *Arduino Mega*, LCD 16x4, modul *reader* MFRC522 dan *Buzzer* seperti skema pada Gambar 4.3 dan agar pengujian dapat berjalan menggunakan program dengan alur seperti dalam *flowchart* Gambar 3.18 dan Gambar 3.19.



Gambar 4.3 Skema untuk Pengujian RFID dengan LCD 16x4 untuk Menampilkan Nama Obyek

Cara pengujian untuk pengambilan data yang digunakan adalah dengan melakukan *scanning* UID *tag* dengan menggunakan modul *reader* MFRC522 yang telah terhubung dengan *Arduino Mega* dalam alat pemutar narasi, kemudian dengan cara mendekatkan *tag* ke alat pemutar narasi obyek dengan jarak 1-3 cm sesuai jarak baca modul *reader* yang telah terdeteksi dengan skema pengujian seperti pada Gambar 4.3. Untuk pengujian menggunakan 5 *tag* yaitu *tag* A, *tag* B, *tag* C, *tag* D dan *tag* E untuk tiap *tag* mewakili tiap obyek. Dengan program yang telah diupload pada *Arduino* sebelumnya, program yang digunakan adalah program menampilkan data UID dengan metode program *dump byte array* dalam bentuk *array* UID yang terdapat pada *tag* RFID. Apabila modul *reader* mendeteksi *tag* maka *buzzer* akan berbunyi dan itu menandakan bahwa *reader* berhasil mendeteksi UID *tag*, sehingga pada serial *monitor* akan tampil nilai UID *tag* dengan nomer UID yang berbeda-beda tiap *tag* , dan tiap UID pada *tag* mewakili nama obyek yang tampil pada layar LCD 16x4. Berikut hasil pengambilan data yang diperoleh ditunjukkan pada Tabel 4.2 – Tabel 4.4.

Tabel 4.2 Hasil Pembacaan UID *Tag* Pada Jarak 1cm

Jarak <i>Tag</i> dari <i>Reader</i> (cm)	<i>Tag</i> RFID	Nomor UID	Nama Obyek yang Tampil pada LCD
1 cm	A	362fd58c	<i>The Scream</i>
1 cm	B	50df1e83	Eiffel
1 cm	C	8a0956ca	<i>La Scapigliata</i>
1 cm	D	aaa565ca	<i>Liberty</i>
1 cm	E	ca0ef4a0	<i>Map Of Hell</i>

Tabel 4.3 Hasil Pembacaan UID *Tag* Pada Jarak 2cm

Jarak <i>Tag</i> dari <i>Reader</i> (cm)	<i>Tag</i> RFID	Nomor UID	Nama Obyek yang Tampil pada LCD
2 cm	A	362fd58c	<i>The Scream</i>
2 cm	B	50df1e83	Eiffel

Jarak Tag dari Reader (cm)	Tag RFID	Nomor UID	Nama Obyek yang Tampil pada LCD
2 cm	C	8a0956ca	<i>La Scapigliata</i>
2 cm	D	aaa565ca	<i>Liberty</i>
2 cm	E	ca0ef4a0	<i>Map Of Hell</i>

Tabel 4.4 Hasil Pembacaan UID Tag Pada Jarak 3cm

Jarak Tag dari Reader (cm)	Tag RFID	Nomor UID	Nama Obyek yang Tampil pada LCD
3 cm	A	362fd58c	<i>The Scream</i>
3 cm	B	50df1e83	<i>Eiffel</i>
3 cm	C	8a0956ca	<i>La Scapigliata</i>
3 cm	D	aaa565ca	<i>Liberty</i>
3 cm	E	ca0ef4a0	<i>Map Of Hell</i>

Dari hasil *scanning* data UID tag yang terdapat pada Tabel 4.2 – 4.4 dapat diketahui bahwa 5 tag dengan kode-kode unik dapat terbaca oleh modul reader MFRC522 hingga jarak 3 cm dan ditampilkan pada LCD 16x4 dalam bentuk nama obyek dengan 5x percobaan. Kode-kode ini masing-masing mewakili satu obyek dalam percobaan ini.

4.3 Pengujian Arduino Mega 2560 dengan MP3 Shield

Pengujian *Arduino Mega 2560* dengan *MP3 shield* merupakan fokus pekerjaan dari Dany Prasetyo. Pengujian *MP3 shield* ini bertujuan agar dapat mengetahui kesesuaian antara UID tag yang terbaca dengan suara penjelasan yang dimainkan. Selain itu dilakukan juga pengamatan terhadap lamanya waktu pemutaran penjelasan tiap obyek ketika MP3 mulai dimainkan. Lamanya waktu pemutaran penjelasan obyek diambil ketika MP3 *shield* berbunyi sampai MP3 *shield* berhenti memainkan suara penjelasan obyek. Pengukuran lamanya waktu penjelasan ini dilakukan dengan *stopwatch*. Pengujian ini didasarkan dengan program untuk memutar penjelasan obyek yang sesuai seperti pada Gambar 4.4.

```

void menu () {
  if (digitalRead(A) == HIGH) {
    if (temp==sample1) {
      lcd.clear();
      lcd.setCursor(-1,2);
      lcd.print("THE SCREAM");
      lcd.setCursor(0,3);
      lcd.print("INDONESIA");
      mp3.play_sd(0x0006);
      EEPROM.write(addr,0006);}

    if (temp==sample2) {
      lcd.clear();
      lcd.setCursor(1,2);
      lcd.print("EIFFEL");
      lcd.setCursor(0,3);
      lcd.print("INDONESIA");
      mp3.play_sd(0x0002);
      EEPROM.write(addr,0002);}

    if (digitalRead(B) == HIGH) {
      if (temp==sample1) {
        lcd.clear();
        lcd.setCursor(-1,2);
        lcd.print("THE SCREAM");
        lcd.setCursor(0,3);
        lcd.print("ENGLISH");
        mp3.play_sd(0x0001);
        EEPROM.write(addr,0001);}

      if (temp==sample2) {
        lcd.clear();
        lcd.setCursor(0,2);
        lcd.print("EIFFEL");
        lcd.setCursor(0,3);
        lcd.print("ENGLISH");
        mp3.play_sd(0x0007);
        EEPROM.write(addr,0007);}
    }
  }
}

```

Gambar 4.4 Listing Program untuk Pengujian Arduino Mega 2560 dengan MP3 Shield

Pada Gambar 4.4 adalah program untuk memanggil obyek yang telah tersimpan pada SD-card MP3 shield dengan perintah MP3 *play*. Terdapat 2 pilihan Bahasa yang digunakan yaitu pada tombol A Bahasa Indonesia sedangkan untuk tombol B untuk memanggil obyek Bahasa English. Penyimpanan data menggunakan kode yang berbeda-beda seperti pada Tabel 4.5 sedangkan hasil pengujian untuk melihat sesuai atau tidaknya data yang dipanggil Arduino dengan alamat obyek pada SD-card MP3 shield dapat dilihat pada Tabel 4.6



Gambar 4.5 Replika Obyek Museum Untuk Pengambilan Data

Cara pengambilan data yang dilakukan adalah dengan mendekatkan alat pemutar narasi ke replika obyek museum seperti pada Gambar 4.5 dengan jarak baca 1-3 cm. *Tag* RFID diletakkan di balik obyek sehingga alat pemutar narasi bisa mendeteksi setiap obyek sesuai obyek yang dipilih, kemudian ketika modul *reader* telah berhasil membaca *UID tag* maka *MP3 shield* akan mulai memainkan narasi sesuai obyek yang terpilih pada Tabel 4.6 adalah hasil pengujian RFID dengan *MP3 shield* untuk mengetahui kesesuaian antara alamat *UID tag* dengan alamat obyek yang telah tersimpan dalam *SD-card* yang diunjukkan pada Tabel 4.5 yaitu terdapat 5 jenis narasi obyek dengan 2 bahasa, Bahasa Indonesia dan Bahasa Inggris dengan kode obyek yang berbeda-beda contoh Menara Eiffel Bahasa Indonesia dengan kode '0002'. Kemudian untuk mengukur lamanya waktu pemutaran dengan menggunakan *stopwatch* ketika *MP3* mulai memainkan narasi obyek sampai *MP3* berhenti.

Tabel 4.5 Data Obyek Pada *SD-card* *MP3 Shield*

No.	Obyek	Bahasa	Kode Obyek
1	Eiffel	Indonesia	0002
2	<i>La scapigliata</i>	Indonesia	0003
3	<i>Liberty</i>	Indonesia	0004

No.	Obyek	Bahasa	Kode Obyek
4	<i>Map Of Hell</i>	Indonesia	0005
5	<i>The Scream</i>	Indonesia	0006
6	<i>The Scream</i>	English	0001
7	Eiffel	English	0007
8	<i>La scapigliata</i>	English	0008
9	<i>Liberty</i>	English	0009
10	<i>Map Of Hell</i>	English	000A

Tabel 4.6 Hasil Pengujian Antara RFID dengan Obyek

No.	Obyek Pada MP3 Shield	UID Tag	Sesuai/ TidakSesuai	Durasi (Menit)
1	Eiffel (Indonesia)	50df1e83	✓	01:32
2	<i>Map Of hell</i> (Indonesia)	ca0ef4a0	✓	01:28
3	<i>Liberty</i> (Indonesia)	aaa565ca	✓	01:36
4	<i>La scapigliata</i> (Indonesia)	8a0956ca	✓	01:35
5	<i>The Scream</i> (Indonesia)	362fd58c	✓	01:55
6	Eiffel (English)	50df1e83	✓	01:08
7	<i>Map Of hell</i> (English)	ca0ef4a0	✓	01:08
8	<i>Liberty</i> (English)	aaa565ca	✓	01:12
9	<i>La scapigliata</i> (English)	8a0956ca	✓	01:15
10	<i>The Scream</i> (English)	362fd58c	✓	01:25

Pada Tabel 4.5 dapat diketahui bahwa setiap narasi obyek disimpan pada *SD-card* yang terdapat pada modul *MP3 shield* dengan kode alamat yang berbeda-beda. Dari hasil pengamatan pada Tabel 4.6 adalah hasil pengujian obyek yang telah berhasil teridentifikasi dengan sistem kerja ketika alat pemutar narasi yang terdapat *reader* MFRC522 didekatkan ke bagian yang terdapat *tag* RFID pada obyek dengan jarak maksimal 3cm maka *buzzer* akan berbunyi, sehingga dapat diketahui bahwa *UID tag* yang terbaca oleh *reader* mampu memanggil data mp3 sesuai dengan alamat kode mp3 yang telah disesuaikan dengan alamat tiap *UID tag* dan juga diperoleh waktu pemutaran penjelasan obyek yang berbeda-beda tergantung dari obyek yang telah teridentifikasi oleh *reader* pada *tag* RFID.

4.4 Pengujian RTC

Pengujian RTC merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Tujuan dari pengujian RTC ini bertujuan untuk melihat apakah RTC sudah bekerja sesuai dengan waktu yang sedang berjalan. Pengujian dilakukan dengan membandingkan waktu yang ditampilkan pada LCD dengan waktu yang berjalan pada laptop. Dalam pengujian dilakukan pengambilan data waktu sebanyak 8 kali dengan menghubungkan rangkaian RTC pada *Arduino Mega* dengan menuliskan program seperti pada Gambar 4.6.

```
void jam ()
{
    //RTC.adjust(DateTime(__DATE__, __TIME__));
    DateTime now = RTC.now();
    lcd.setCursor(8, 0);
    lcd.print(now.hour(), DEC);
    lcd.print(":");
    lcd.print(now.minute(), DEC);
    lcd.print(":");
    lcd.print(now.second(), DEC);
    lcd.print(" ");

    Serial.println();
}
```

Gambar 4.6 Program Void jam

Hasil pengambilan data untuk pengujian RTC ini yang ditunjukkan pada Tabel 4.7 dengan cara melihat waktu yang ditampilkan pada LCD yang terletak pada alat dan melihat waktu yang berada pada laptop. Waktu yang dilihat pada alat dan waktu pada laptop dibandingkan untuk mengetahui apakah waktu yang digunakan pada alat sudah sesuai dengan waktu yang sedang berjalan dengan waktu pada laptop digunakan sebagai acuan.

Tabel 4.7 Hasil Pengujian Waktu RTC dengan Waktu pada Laptop

No.	Waktu yang Tampil pada LCD	Waktu pada Laptop
1.	21:6:14	21:06
2.	21:10:8	21:10
3.	21:14:18	21:14
4.	21:36:20	21:36
5.	21:45:3	21:45
6.	22:8:5	22:08
7.	22:17:4	22:17
8.	22:26:21	22:26

Dari Tabel 4.7 yang merupakan pengujian waktu RTC tersebut dapat dilihat bahwa waktu yang ditampilkan pada LCD sesuai dengan waktu yang ditunjukkan pada laptop. Berdasarkan data yang diambil selama 10 kali, waktu yang ditampilkan pada LCD sama dengan waktu yang ditunjukkan pada laptop. Dengan hasil dari data yang telah diambil, presentase kesalahannya adalah 0% dan dapat dikatakan bahwa RTC telah bekerja dengan baik.

4.5 Pengujian Komunikasi Serial Arduino dengan PC

Pengujian komunikasi serial ini merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Pengujian komunikasi serial antara *Arduino* dengan PC ini bertujuan untuk mengetahui apakah komunikasi serial antara *Arduino* dengan PC dapat dilakukan untuk nantinya dimasukkan pada antarmuka *Visual Basic* atau VB. Komunikasi serial dikatakan berjalan dengan baik apabila data yang dikirimkan dari

Arduino dapat diterima oleh PC dengan data yang sama dengan *listing* program seperti pada Gambar 4.7.

```

if (!Serial.available()) {
    address=addr;
    value=EEPROM.read(address);
    Serial.print('0'),Serial.print(value);
    Serial.println();

    address=address+1;
    if (address==512)
        address=0;

    addr=addr+1;
    if (addr==512)
        addr=0;

    delay(300);
}
}

```

Gambar 4.7 *Listing* Program untuk Pengiriman Data dari EEPROM *Arduino* untuk Komunikasi Serial

Cara untuk melakukan pengujian ini dilakukan dengan mengirimkan data dari *Arduino* ke PC dengan *baud rate* yang diatur dan data yang dikirim akan dibandingkan dengan data yang diterima. Langkah pertama adalah dengan mengatur *baud rate* pada *Arduino* sebelum alat dihubungkan dengan PC. Ketika alat sudah dihubungkan ke PC, untuk melihat data yang diterima di PC menggunakan serial monitor pada *software Arduino* dengan *setting baud rate* yang sesuai dengan program pada *Arduino*. Komunikasi serial ini dapat berjalan dengan potongan program seperti pada Gambar 4.6 yang mengambil data dari EEPROM *Arduino* untuk dapat dikirimkan ke PC melalui komunikasi serial. Untuk hasil dari pengujian ini dapat dilihat pada Tabel 4.8

Tabel 4.8 Pengujian Komunikasi Serial *Arduino* dengan PC

No.	<i>Baud Rate</i>	Data yang Dikirim	Data yang Diterima
1.	4800	0001	01
2.	4800	0002	02
3.	4800	0003	03

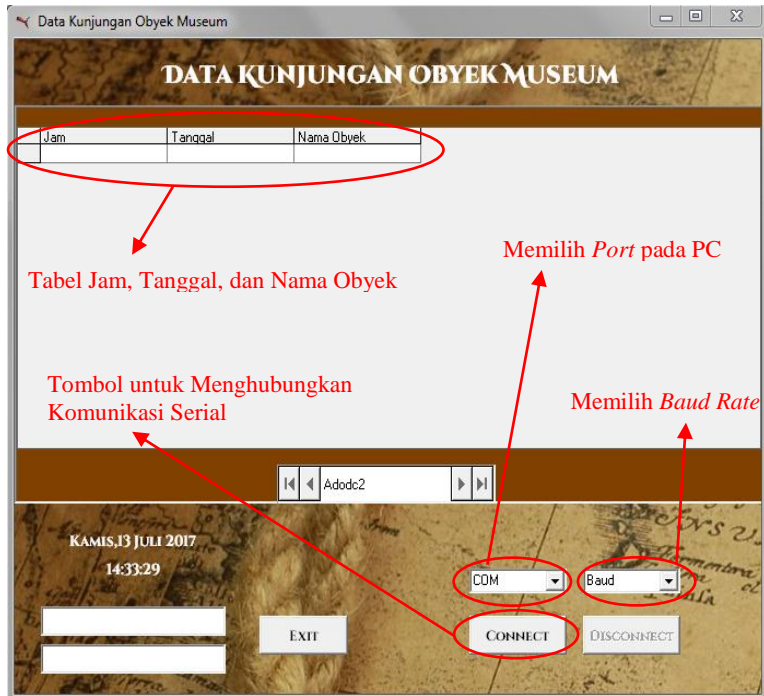
No.	<i>Baud Rate</i>	Data yang Dikirim	Data yang Diterima
4.	4800	0004	04
5.	4800	0005	05
6.	9600	0001	01
7.	9600	0002	02
8.	9600	0003	03
9.	9600	0004	04
10.	9600	0005	05
11.	19200	0001	01
12.	19200	0002	02
13.	19200	0003	03
14.	19200	0004	04
15.	19200	0005	05
16.	38400	0001	01
17.	38400	0002	02
18.	38400	0003	03
19.	38400	0004	04
20.	38400	0005	05

Berdasarkan data yang diambil di Tabel 4.8 sebanyak 5 kali untuk setiap nilai *baud rate*, dengan *range* nilai *baud rate* antara 4800-38400. Pada setiap nilai *baud rate*, data yang muncul pada *serial monitor* adalah angka 01-05 dengan data yang dikirim adalah angka 0001-0005. Dari hasil pengujian tersebut dapat dikatakan bahwa komunikasi serial dapat berjalan pada *range* nilai *baud rate* antara 4800-38400 karena terdapat kesesuaian pada dua digit terakhir dari data yang dikirim dan diterima.

4.6 Pengujian Antarmuka *Visual Basic*

Pengujian antarmuka *Visual Basic* merupakan fokus pekerjaan dari Cahyo Dhian Tyastono. Pengujian dari antarmuka bertujuan untuk melihat apakah antarmuka sudah tersambung dengan *Arduino* dan bisa menampilkan data kunjungan obyek museum. Pada tampilan awal ketika antarmuka dijalankan seperti pada Gambar 4.8 belum ada data yang masuk sehingga tabel pada *Datagrid* kosong dan komunikasi serial

belum terhubung dan COM serta *baud rate* belum diatur untuk komunikasi serial. Pada antarmuka *Visual Basic* selain tampilan pada *form designer* terdapat program agar antarmuka dapat berjalan sesuai dengan yang diharapkan.



Gambar 4.8 Tampilan Awal Aplikasi Antarmuka Data Obyek Museum

Cara untuk pengujian antarmuka *Visual Basic* pada mulanya adalah dengan menghubungkan komunikasi serial antara *Arduino* dimana pada antarmuka *Visual Basic* diperlukan pengaturan *port* yang digunakan dengan memilih pada *ComboBox* COM dan pilih sesuai dengan yang digunakan untuk komunikasi serial, dengan COM 9 untuk komputer yang digunakan untuk pengujian. Setelah itu pilih juga *baud rate* yang digunakan pada *ComboBox* *baud rate*, untuk pengujian ini nilai *baud rate* yang digunakan adalah 9600. Kemudian tekan tombol *CONNECT* untuk menyambungkan komunikasi serial. Setelah tombol *CONNECT* ditekan, maka pada tabel pada *Datagrid* akan muncul data kunjungan

obyek museum sesuai dengan data yang diambil dari *Arduino*. *Set-up* untuk komunikasi serial pada *Visual Basic* berdasarkan dari program *Visual Basic* yang alurnya seperti dalam *flowchart* pada Gambar 3.21. Agar antarmuka dapat menampilkan nama obyek sesuai dengan data yang dikirim dari *Arduino* melalui komunikasi serial didasarkan dari program pada *Visual Basic* yang alurnya seperti dalam *flowchart* pada Gambar 3.22.

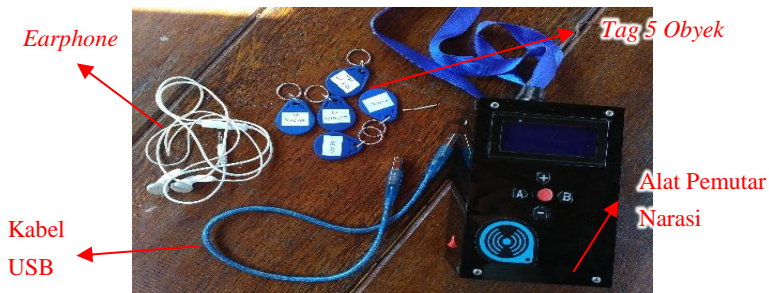


Gambar 4.9 Tampilan Aplikasi Antarmuka Setelah Komunikasi Disambungkan

Berdasarkan hasil pada antarmuka seperti pada Gambar 4.9 terlihat bahwa data yang dikirim dari *Arduino* telah masuk ke PC dan antarmuka *Visual Basic* dapat menampilkan nama obyek museum sesuai data yang telah masuk. Dengan hasil tersebut maka dapat dikatakan bahwa aplikasi antarmuka ini dapat berjalan dengan baik.

4.7 Pengujian Keseluruhan Alat

Pengujian keseluruhan alat merupakan fokus pekerjaan Bersama. Pengujian ini meliputi keseluruhan blok sistem, dimana pengujian ini bertujuan untuk mengetahui apakah sistem dapat berkerja sesuai dengan apa yang diencanakan, yaitu RFID dapat mengidentifikasi pemutaran penjelasan obyek museum secara otomatis. Pengujian dilakukan sebanyak 25 kali dengan masing-masing pengujian 5 kali per obyek. Adapun alat pengujian dapat dilihat pada Gambar 4.10 dengan program untuk menjalankan alat seperti pada Lampiran B nomor 1 yang merupakan *listing* program *Arduino*.



Gambar 4.10 Pengujian Alat Pemutar Narasi

Cara pengujian alat pemutar narasi obyek secara keseluruhan yang terdiri dari 5 tag RFID yaitu menggunakan tag A, tag B, tag C, tag D, dan tag E untuk pengujian tiap obyek dengan menggunakan earphone, kabel serial USB, dan alat pemutar narasi yang terdapat pada Gambar 4.10. Untuk memperoleh hasil pengujian sesuai pada Tabel 4.9 yaitu pada tag A yang diletakkan pada obyek Menara Eiiifel sehingga nantinya suara penjelasan yang keluar adalah penjelasan obyek tentang Menara Eiiifel. Untuk cara pengambilan data pada obyek tag A sama seperti skema pada Gambar 4.3. Cara menggunakan alat ini dengan menekan saklar dalam kondisi *on* pada alat kemudian akan tampil perintah dekatkan alat pada layar LCD 16x4 sebagai petunjuk penggunaan alat kemudian dekatkan alat ke replika museum seperti pada Gambar 4.5 maka akan tampil pilihan Bahasa, kemudian pilih bahasa yang diinginkan maka alat pemutar narasi secara otomatis akan memainkan penjelasan obyek sesuai obyek yang telah dipilih dalam bentuk suara audio selama 1 menit 32 detik dengan waktu respon 1 detik dan dengan 5 Kali percobaan sukses . Adapun cara penggunaan dan hasil pengujian dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian Keseluruhan Sistem Alat Pemutar Narasi Obyek

<i>Reader Didekatkan pada Kartu</i>	<i>Tampilan LCD</i>	<i>Respon MP3 Shield</i>	<i>Waktu Respon</i>	<i>Hasil Pengujian</i>	
				<i>Sukses</i>	<i>Gagal</i>
A	<ul style="list-style-type: none"> • <i>Welcome</i> • Dekatkan Alat • Pilih Bahasa • Obyek Menara Eiffel Indonesia 	Memutar suara penjelasan 01:32 menit	1 detik	5 Kali	-
B	<ul style="list-style-type: none"> • <i>Welcome</i> • Dekatkan Alat • Pilih Bahasa • Obyek Map Of hell Indonesia 	Memutar suara penjelasan 01:28 menit	1 detik	5 Kali	-
C	<ul style="list-style-type: none"> • <i>Welcome</i> • Dekatkan Alat • Pilih Bahasa • Obyek Liberty English 	Memutar suara penjelasa 01:12 menit	1 detik	5 Kali	-
D	<ul style="list-style-type: none"> • <i>Welcome</i> • Dekatkan Alat • Pilih Bahasa • Obyek La scapiagliata English 	Memutar suara penjelasa 01:15 menit	1 detik	5 Kali	-
E	<ul style="list-style-type: none"> • <i>Welcome</i> • Dekatkan Alat • Pilih Bahasa • Obyek The Scream English 	Memutar suara penjelasa 01:25 menit	1 detik	5 Kali	-

Dari hasil pengujian pada Tabel 4.9 didapatkan perhitungan presentase kesalahan eksekusi sebagai berikut:

%Kesalahan

$$\begin{aligned}\text{Eksekusi} &= \left| \frac{\text{Total percobaan} - \text{percobaan berhasil}}{\text{total percobaan}} \right| \times 100\% \\ &= \left| \frac{25 - 25}{25} \right| \times 100\% \\ &= 0\%\end{aligned}$$

Jadi presentase kesalahan eksekusi data adalah 0% dari keseluruhan pengujian alat.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari pengujian terhadap sistem dapat diambil beberapa kesimpulan sebagai berikut:

1. Jarak pembacaan modul *reader* MFRC522 terhadap *tag* maksimal 3 cm.
2. Modul *reader* MFRC522 bisa diintegrasikan dengan modul MP3 *shield* dengan hasil yang optimal pada sistem identifikasi narasi obyek.
3. Suara Narasi obyek museum yang digunakan mempunyai durasi yang berbeda-beda dengan durasi yang paling pendek selama 1 menit 8 detik dan durasi paling lama 1 menit 55 detik.
4. RTC dapat menampilkan waktu yang tepat yang dapat dilihat dari perbandingannya dengan waktu yang ditunjukkan pada laptop yang menunjukkan tidak ada perbedaan waktu.
5. *Arduino* dan PC dapat melakukan komunikasi serial dan data dapat diterima dengan nilai *baud rate* 9600 pada aplikasi basis-data *Visual Basic* dengan kesesuaian pada dua digit terakhir pada data yang dikirim dan diterima dan ditampilkan pada antarmuka.

Dari pembuatan tugas akhir ini masih terdapat beberapa kekurangan pada alat serta sistemnya sehingga perlu evaluasi untuk ke depannya dengan beberapa saran sebagai berikut:

1. Alat ini masih berupa *prototype* dengan dimensi yang cukup besar dan untuk ke depannya bias lebih dipraktiskan agar dimensi lebih kecil dan lebih nyaman saat dibawa pengunjung.
2. *Reader* yang digunakan kemampuannya masih kurang bagus dan untuk ke depannya dapat menggunakan *reader* yang lebih baik dalam pembacaan jarak.
3. Akan atau lebih baik lagi jika data kunjungan obyek museum dapat dilihat secara *real time* dengan komunikasi jarak jauh sehingga data tidak perlu dimasukkan dengan menyambungkan alat dengan komputer melalui kabel melainkan data bisa langsung dilihat ketika alat digunakan pada obyek museum.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

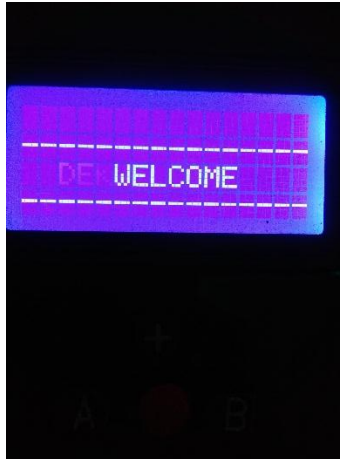
- [1] Sriyanto, Eko. Sistem Akses Kontrol Menggunakan Radio Frekuensi Identification (RFID). **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Surabaya: 2006.
- [2] Respito, Cahyo. Perancangan dan Pembuatan Perangkat Lunak Otomasi dan Keamanan Asset Tracking dengan Memanfaatkan Teknologi RFID. **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Surabaya: 2007.
- [3] Hanung, Charlos. Penghitung Laju Menggunakan RFID Berbasis Arduino. **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Sanata Dharma Yogyakarta. Yogyakarta: 2016.
- [4] Himawan, Hendra. Perancangan Jam Digital Berbasis Mikrokontroler AT8535 dan RTC DS1307. **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Mercu Buana. Jakarta: 2014
- [5] Jati, Agung Nugroho, dkk., Performance Analysis of NFRC522 RFID Reader on Table Location Information System In Food Court, **Jurnal e-Proceeding of Engineering**, Jurusan Sistem Informasi, Fakultas Teknik Elektro, Universitas Telkom. Bandung: 2015.
- [6] Rawashdeh, Mohannad. **Arduino MP3 Shield**, URL: <http://www.instructables.com/id/Arduino-MP3-Shield/?ALLSTEPS>, Di akses pada tanggal 5 Mei 2015.
- [7] Lubis, Reza Al Kautsar. Perancangan Sistem Verifikasi Keanggotaan dengan Kartu Cerdas Nirkontak Berbasis Arduino Mega 2560. **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sumatera Utara. Medan: 2016.
- [8] Supardi, Ir. Yuniar. **Semua Bisa Menjadi Programmer VB 6 Hingga VB 2008 Basic**, PT. Elex Media Komputindo, Jakarta:2011
- [9] _____, **Renaissance – Victoria and Albert Museum**, URL: <http://www.vam.co.uk/page/r/renaissance>, Diakses pada tanggal 18 Mei 2017.

-----Halaman ini sengaja dikosongkan-----

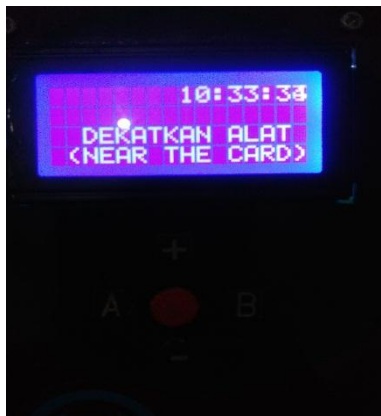
LAMPIRAN

A. Dokumentasi Alat

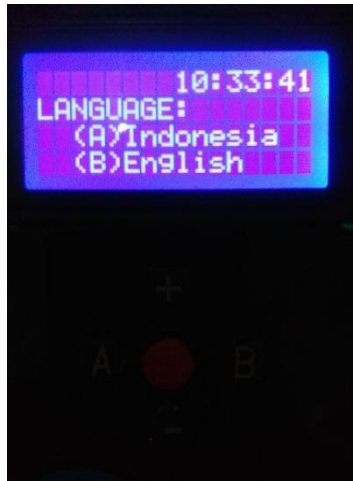
1. Tampilan Awal pada LCD



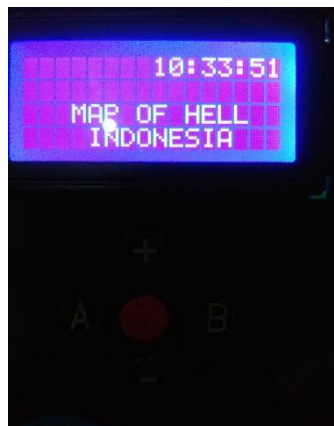
2. Tampilan Perintah untuk mendeteksi *tag* RFID



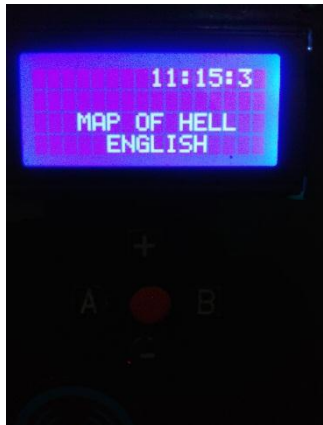
3. Tampilan pada LCD untuk Memilih Bahasa



4. Tampilan Obyek dengan Bahasa Indonesia pada LCD yang telah teridentifikasi



5. Tampilan Obyek dengan Bahasa Indonesia pada LCD yang telah teridentifikasi



-----Halaman ini sengaja dikosongkan-----

B. Listing Program

1. Listing Program Arduino

```

/*
 * Typical pin layout used:
 * -----
 *
 * MFRC522  Arduino  Arduino  Arduino  Arduino
Arduino
 *          Reader/PCD  Uno          Mega      Nano v3
Leonardo/Micro Pro Micro
 * Signal  Pin    Pin    Pin    Pin    Pin    Pin
 * -----
 *
 * RST/Reset  RST      9      5      D9      RESET/ICSP-5
RST
 * SPI SS     SDA(SS)  10      53      D10      10
10
 * SPI MOSI   MOSI     11 / ICSP-4  51      D11      ICSP-4
16
 * SPI MISO   MISO     12 / ICSP-1  50      D12      ICSP-1
14
 * SPI SCK    SCK      13 / ICSP-3  52      D13      ICSP-3
15
 */
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <MP3.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include "RTCLib.h"
#define SS_PIN 53
#define RST_PIN 44

LiquidCrystal_I2C lcd(0x3F,16,4);
MFRC522 mfrc522(SS_PIN, RST_PIN);

```

```

MFRC522::MIFARE_Key key;
MP3 mp3;
SoftwareSerial Geno(7,8); // Rx , Tx
RTC_DS1307 RTC;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};
String temp="";
String sample1="362fd58c";
String sample2="50df1e83";
String sample3="8a0956ca";
String sample4="aaa565ca";
String sample5="ca0ef4a0";
int A = 22;
int B = 23;
int stopp = 26;
int vol2=24;
int vol1=25;
int kondisi = 0;
int up=20;
int buzzer = 45;
byte value;
int address=0;
int addr=0;

void setup()
{
    pinMode(A, INPUT);
    pinMode(B, INPUT);
    pinMode(stopp, INPUT);
    pinMode(vol1, INPUT);
    pinMode(vol2, INPUT);
    pinMode(buzzer, OUTPUT);
    Wire.begin();
    Serial.begin(9600);
    SPI.begin();
    RTC.begin();
    lcd.begin();
    lcd.backlight();

```



```

    mfrc522.PCD_Init(); // Init MFRC522 card
    mp3.begin(MP3_SOFTWARE_SERIAL);
    lcd.setCursor(0,1);
    lcd.print("-----");
    lcd.setCursor(0,2);
    lcd.print(" WELCOME ");
    lcd.setCursor(-4,3);
    lcd.print("-----");
    delay(1000);
    lcd.clear();
    lcd.setCursor(-2,2);
    lcd.print("DEKATKAN ALAT");
    lcd.setCursor(-3,3);
    lcd.print("(NEAR THE CARD)");
}

void jam ()
{
    //RTC.adjust(DateTime(__DATE__, __TIME__));
    DateTime now = RTC.now();
    lcd.setCursor(8, 0);
    lcd.print(now.hour(), DEC);
    lcd.print(":");
    lcd.print(now.minute(), DEC);
    lcd.print(":");
    lcd.print(now.second(), DEC);
    lcd.print(" ");

    Serial.println();
}

void RFID()
{
    if ( ! mfrc522.PICC_IsNewCardPresent())
    {
        return;
    }
    if ( ! mfrc522.PICC_ReadCardSerial())
    {

```

```

        return;
    }

    dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
    lcd.clear();
    lcd.setCursor(0,2);
    lcd.println(temp);
    lcd.setCursor(0,1);
    lcd.print("LANGUAGE:");
    lcd.setCursor(-2,2);
    lcd.print("(A)Indonesia");
    lcd.setCursor(-2,3);
    lcd.print("(B)English");
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    mfrc522.PICC_HaltA();
    mfrc522.PCD_StopCrypto1();
}

void volum()
{
    if(digitalRead(vol2) == 1)
    {
        up=up;
        kondisi=0;
        mp3.volume (up);
    }
    if(digitalRead(vol2) == 0 && kondisi ==0)
    {
        up=up+1;
        kondisi=1;
        mp3.volume (up);
    }
    if(digitalRead(vol2) == 0 && kondisi==1)
    {
        up=up;
        kondisi=1;
        mp3.volume (up);
    }
}

```

```

    if(up>=30){
        up=30;
        kondisi=1;
        mp3.volume (up);
    }
}
if(digitalRead(vol1) == 1)
{
    up=up;
    kondisi=0;
    mp3.volume (up);
}
if(digitalRead(vol1) == 0 && kondisi ==0)
{
    up=up-1;
    kondisi=1;
    mp3.volume (up);
}
if(digitalRead(vol1) == 0 && kondisi==1)
{
    up=up;
    kondisi=1;
    mp3.volume (up);
    if(up<=15){
        up=15;
        kondisi=1;
        mp3.volume (up);
    }
}
}

void menu ()
{
    if(digitalRead(A) == HIGH)
    {
        if(temp==sample1){
            lcd.clear();
            lcd.setCursor(-1,2);
            lcd.print("THE SCREAM");
        }
    }
}

```

```

        lcd.setCursor(0,3);
        lcd.print("INDONESIA");
        mp3.play_sd(0x0006);
        EEPROM.write(addr,0006);
    }
    if(temp==sample2){
        lcd.clear();
        lcd.setCursor(1,2);
        lcd.print("EIFFEL");
        lcd.setCursor(0,3);
        lcd.print("INDONESIA");
        mp3.play_sd(0x0002);
        EEPROM.write(addr,0002);
    }
    if(temp==sample3){
        lcd.clear();
        lcd.setCursor(-3,2);
        lcd.print("LA SCAPIGLIATA");
        lcd.setCursor(0,3);
        lcd.print("INDONESIA");
        mp3.play_sd(0x0003);
        EEPROM.write(addr,0003);
    }
    if(temp==sample4){
        lcd.clear();
        lcd.setCursor(1,2);
        lcd.print("LIBERTY");
        lcd.setCursor(0,3);
        lcd.print("INDONESIA");
        mp3.play_sd(0x0004);
        EEPROM.write(addr,0004);
    }
    if(temp==sample5){
        lcd.clear();
        lcd.setCursor(-1,2);
        lcd.print("MAP OF HELL");
        lcd.setCursor(0,3);
        lcd.print("INDONESIA");
        mp3.play_sd(0x0005);
    }

```

```

        EEPROM.write(addr,0005);

    }
}

if(digitalRead(B) == HIGH)
{
    if(temp==sample1){
        lcd.clear();
        lcd.setCursor(-1,2);
        lcd.print("THE SCREAM");
        lcd.setCursor(0,3);
        lcd.print("ENGLISH");
        mp3.play_sd(0x0001);
        EEPROM.write(addr,0001);
    }
    if(temp==sample2){
        lcd.clear();
        lcd.setCursor(0,2);
        lcd.print("EIFFEL");
        lcd.setCursor(0,3);
        lcd.print("ENGLISH");
        mp3.play_sd(0x0007);
        EEPROM.write(addr,0007);
    }
    if(temp==sample3){
        lcd.clear();
        lcd.setCursor(-3,2);
        lcd.print("LA SCAPIGLIATA");
        lcd.setCursor(0,3);
        lcd.print("ENGLISH");
        mp3.play_sd(0x0008);
        EEPROM.write(addr,0012);
    }
    if(temp==sample4){
        lcd.clear();
        lcd.setCursor(0,2);
        lcd.print("LIBERTY");
        lcd.setCursor(0,3);

```

```

        lcd.print("ENGLISH");
        mp3.play_sd(0x0009);
        EEPROM.write(addr,0011);

    }
    if(temp==sample5){
        lcd.clear();
        lcd.setCursor(-1,2);
        lcd.print("MAP OF HELL");
        lcd.setCursor(1,3);
        lcd.print("ENGLISH");
        mp3.play_sd(0x000A);
        EEPROM.write(addr,0010);
    }
}

if(digitalRead(stopp) == HIGH)
{
    mp3.stop();
}

if (!Serial.available()){
    address=addr;
    value=EEPROM.read(address);
    Serial.print('O'),Serial.print(value);
    Serial.println();

    address=address+1;
    if(address==512)
        address=0;

    addr=addr+1;
    if(addr==512)
        addr=0;

    delay(500);
}

}

void loop()
{

```

```

menu();
RFID();
volum();
jam();
}

void dump_byte_array (byte *buffer, byte bufferSize) {
    temp="";
    for (byte i = 0; i < bufferSize; i++) {

        temp += String(buffer[i] < 0x10 ? "0" : "");
        temp += String(buffer[i], HEX);
    }
}

```

2. Listing Program Visual Basic

```

Dim Conn1 As ADODB.Connection
Dim Cmd1 As ADODB.Command
Dim Param1 As ADODB.Parameter
Dim Rs1 As ADODB.Recordset
Dim sHari As String
Dim aHari
Dim nilai As Integer
Dim simpan As Integer

```

```

Private Sub cmdConnect_Click()
Dim port As Integer
On Error GoTo errcode

```

```

If MSComm1.PortOpen = False Then
    MSComm1.CommPort = Combo1.ListIndex + 1
    MSComm1.RThreshold = 1
    MSComm1.InputLen = 0
    MSComm1.Settings = Combo2.List(Combo2.ListIndex) &
",N,8,1"
    MSComm1.PortOpen = True

```

```
cmdConnect.Enabled = False
cmdDisconnect.Enabled = True
Timer4.Enabled = True
Timer2.Enabled = True
Timer3.Enabled = True
End If
```

```
Exit Sub
```

```
errcode:
```

```
MsgBox "Port Salah !", vbOKOnly, "Peringatan"
Combo1.SetFocus
```

```
End Sub
```

```
Private Sub cmdDisconnect_Click()
If MSComm1.PortOpen = True Then
    MSComm1.PortOpen = False
End If
cmdConnect.Enabled = True
cmdDisconnect.Enabled = False
```

```
End Sub
```

```
Private Sub Command1_Click()
Unload Me
End Sub
```

```
Private Sub Form_Load()
aHari = Array("Minggu", "Senin", "Selasa", "Rabu", "Kamis",
"Jumat", "Sabtu")
Dim i As Byte
For i = 1 To 16
Combo1.AddItem (i)
Next i
With Combo2
.AddItem "2400"
.AddItem "4800"
```



```
.AddItem "9600"  
.AddItem "19200"  
.AddItem "38400"  
.AddItem "56600"
```

End With

```
cmdConnect.Enabled = True  
cmdDisconnect.Enabled = False  
End Sub
```

```
Private Sub Timer1_Timer()  
Adodc2.Refresh  
End Sub
```

```
Private Sub Timer3_Timer()  
Dim data As String  
Dim buffer As String
```

```
    If MSComm1.InBufferCount > 0 Then  
        tbkode.Text = ""  
        tbject.Text = ""
```

```
        buffer = MSComm1.Input  
        data = Mid$(buffer, 4, 5)  
        nilai = Val(data)
```

```
        tbkode.Text = nilai
```

```
    If nilai = simpan Then  
        tbject.Text = ""  
    End If
```

```
    If nilai <> simpan Then
```

```
        If nilai = 3 Then
```

```
tbject.Text = "LA SCAPIGLIATA INDO"  
simpan = 3
```

```
End If
```

```
If nilai = 10 Then  
tbject.Text = "LA SCAPIGLIATA ENGLISH"  
simpan = 10
```

```
End If
```

```
If nilai = 2 Then  
tbject.Text = "EIFFEL INDO"  
simpan = 2
```

```
End If
```

```
If nilai = 7 Then  
tbject.Text = "EIFFEL ENGLISH"  
simpan = 7
```

```
End If
```

```
If nilai = 4 Then  
tbject.Text = "LIBERTY INDO"  
simpan = 4
```

```
End If
```

```
If nilai = 9 Then  
tbject.Text = "LIBERTY ENGLISH"  
simpan = 9
```

```
End If
```

```
If nilai = 6 Then  
tbject.Text = "THE SCREAM INDO"  
simpan = 6
```

End If

If nilai = 1 Then
tobject.Text = "THE SCREAM ENGLISH"
simpan = 1
End If

If nilai = 5 Then
tobject.Text = "MAP OF HELL INDO"
simpan = 5
End If

If nilai = 8 Then
tobject.Text = "MAP OF HELL ENGLISH"
simpan = 8
End If

End If

End If

End Sub

Private Sub Timer4_Timer()
Dim SQL As String

Call dbConnect

If tobject.Text <> "" Then

SQL = "INSERT INTO Table1 VALUES('" &
lbljam.Caption & "','" & lbltanggal.Caption & "','" & tobject.Text &
"')"

```

    Dim Rs As Recordset
    Set Rs = New ADODB.Recordset
    Rs.Open SQL, Conn, adOpenDynamic
    tobject.Text = ""

End If

End Sub

Private Sub Timer5_Timer()
    sHari = aHari(Abs(Weekday(Date) - 1))
    lbltanggal.Caption = "" & sHari & ", " & Format(Date, "dd
mmmm yyyy")
    lbljam.Caption = Format(Time, "hh:mm:ss")
End Sub

```

C. Datasheet / Specification

1. Arduino Mega 2560



Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.



www.robotshop.com

La robotique à votre service! - Robotics at your service!



The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH



www.robotshop.com

La robotique à votre service! - Robotics at your service!



value, the LED is on, when the pin is LOW, it's off.

- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I²C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I²C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It



communicates using the original STK500 protocol ([reference](#), [C header files](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

2. MFR522

The following host interfaces are provided:

- Serial Peripheral Interface (SPI)
- Serial UART (similar to RS232 with voltage levels dependant on pin voltage supply)
- I²C-bus interface

3. Features and benefits

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers for connecting an antenna with the minimum number of external components
- Supports ISO/IEC 14443 A/MIFARE and NTAG
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- Supports MF1xxS20, MF1xxS70 and MF1xxS50 encryption in Read/Write mode
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd
- Supports MF1N/MF0UT
- Additional internal power supply to the smart card IC connected via MF1N/MF0UT
- Supported host interfaces
 - ◆ SPI up to 10 Mbit/s
 - ◆ I²C-bus interface up to 400 kBd in Fast mode, up to 3400 kBd in High-speed mode
 - ◆ RS232 Serial UART up to 1228.8 kBd, with voltage levels dependant on pin voltage supply
- FIFO buffer handles 64 byte send and receive
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- Internal oscillator for connection to 27.12 MHz quartz crystal
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test

4. Quick reference data

Table 1. Quick reference data

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V _{DDA}	analog supply voltage	V _{DD(PVDD)} ≤ V _{DDA} = V _{DDD} = V _{DD(TVDD)}	[1][2]	2.5	3.3	3.6	V
V _{DDD}	digital supply voltage	V _{SSA} = V _{SSD} = V _{SS(PVSS)} = V _{SS(TVSS)} = 0 V		2.5	3.3	3.6	V
V _{DD(TVDD)}	TVDD supply voltage			2.5	3.3	3.6	V
V _{DD(PVDD)}	PVDD supply voltage		[2]	1.6	1.8	3.6	V
V _{DD(SVDD)}	SVDD supply voltage	V _{SSA} = V _{SSD} = V _{SS(PVSS)} = V _{SS(TVSS)} = 0 V		1.6	-	3.6	V

Table 1. Quick reference data ...continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{pd}	power-down current	$V_{DDA} = V_{DDD} = V_{DD(TVDD)} = V_{DD(PVDD)} = 3\text{ V}$					
		hard power-down; pin NRSTPD set LOW	[4]	-	5	μA	
		soft power-down; RF level detector on	[4]	-	10	μA	
I_{DD}	digital supply current	pin DVDD; $V_{DDD} = 3\text{ V}$		-	6.5	9	mA
I_{DDA}	analog supply current	pin AVDD; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 0		-	7	10	mA
		pin AVDD; receiver switched off; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 1		-	3	5	mA
$I_{DD(PVDD)}$	PVDD supply current	pin PVDD	[5]	-	-	40	mA
$I_{DD(TVDD)}$	TVDD supply current	pin TVDD; continuous wave	[6][7][8]	-	60	100	mA
T_{amb}	ambient temperature	HVQFN32		-25	-	+85	$^{\circ}\text{C}$

[1] Supply voltages below 3 V reduce the performance in, for example, the achievable operating distance.

[2] V_{DDA} , V_{DDD} and $V_{DD(TVDD)}$ must always be the same voltage.

[3] $V_{DD(PVDD)}$ must always be the same or lower voltage than V_{DDD} .

[4] I_{pd} is the total current for all supplies.

[5] $I_{DD(PVDD)}$ depends on the overall load at the digital pins.

[6] $I_{DD(TVDD)}$ depends on $V_{DD(TVDD)}$ and the external circuit connected to pins TX1 and TX2.

[7] During typical circuit operation, the overall current is below 100 mA.

[8] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between pins TX1 and TX2 at 13.56 MHz.

5. Ordering information

Table 2. Ordering information

Type number	Package		
	Name	Description	Version
MFRC52201HN1/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52201HN1/TRAYBM[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52202HN1/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52202HN1/TRAYBM[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1

[1] Delivered in one tray.

[2] Delivered in five trays.

6. Block diagram

The analog interface handles the modulation and demodulation of the analog signals.

The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa.

Various host interfaces are implemented to meet different customer requirements.

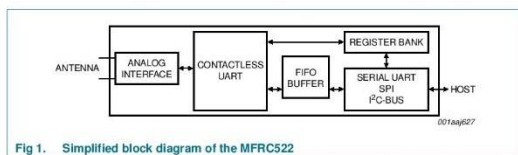
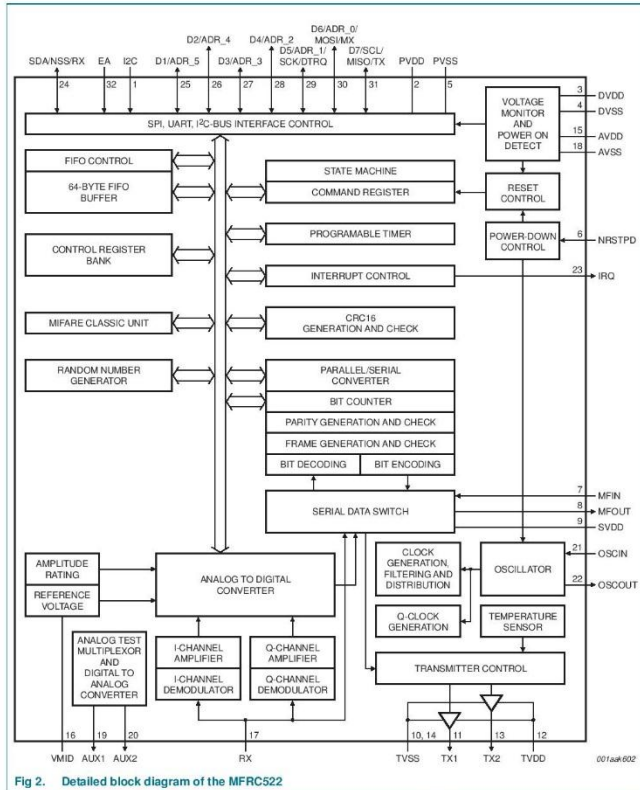


Fig 1. Simplified block diagram of the MFRC522



7. Pinning information

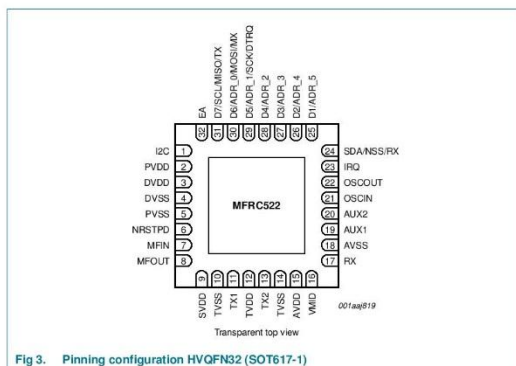


Fig 3. Pinning configuration HVQFN32 (SOT617-1)

7.1 Pin description

Table 3. Pin description

Pin	Symbol	Type ^[1]	Description
1	I2C	I	I ² C-bus enable input ^[2]
2	PVDD	P	pin power supply
3	DVDD	P	digital power supply
4	DVSS	G	digital ground ^[3]
5	PVSS	G	pin power supply ground
6	NRSTPD	I	reset and power-down input: power-down: enabled when LOW; internal current sinks are switched off, the oscillator is inhibited and the input pins are disconnected from the outside world reset: enabled by a positive edge
7	MFIN	I	MIFARE signal input
8	MFOUT	O	MIFARE signal output
9	SVDD	P	MFIN and MFOUT pin power supply
10	TVSS	G	transmitter output stage 1 ground
11	TX1	O	transmitter 1 modulated 13.56 MHz energy carrier output
12	TXDD	P	transmitter power supply: supplies the output stage of transmitters 1 and 2
13	TX2	O	transmitter 2 modulated 13.56 MHz energy carrier output
14	TVSS	G	transmitter output stage 2 ground
15	AVDD	P	analog power supply

MFRC522

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2016. All rights reserved.

Product data sheet
COMPANY PUBLIC

Rev. 3.9 — 27 April 2016
112139

6 of 95

Table 3. Pin description ...continued

Pin	Symbol	Type ^[1]	Description
16	VMID	P	internal reference voltage
17	RX	I	RF signal input
18	AVSS	G	analog ground
19	AUX1	O	auxiliary outputs for test purposes
20	AUX2	O	auxiliary outputs for test purposes
21	OSCIN	I	crystal oscillator inverting amplifier input; also the input for an externally generated clock ($f_{clk} = 27.12$ MHz)
22	OSCOU	O	crystal oscillator inverting amplifier output
23	IRQ	O	interrupt request output: indicates an interrupt event
24	SDA	I/O	I ² C-bus serial data line input/output ^[2]
	NSS	I	SPI signal input ^[2]
	RX	I	UART address input ^[2]
25	D1	I/O	test port ^[2]
	ADR_5	I/O	I ² C-bus address 5 input ^[2]
26	D2	I/O	test port
	ADR_4	I	I ² C-bus address 4 input ^[2]
27	D3	I/O	test port
	ADR_3	I	I ² C-bus address 3 input ^[2]
28	D4	I/O	test port
	ADR_2	I	I ² C-bus address 2 input ^[2]
29	D5	I/O	test port
	ADR_1	I	I ² C-bus address 1 input ^[2]
	SCK	I	SPI serial clock input ^[2]
	DTRQ	O	UART request to send output to microcontroller ^[2]
30	D6	I/O	test port
	ADR_0	I	I ² C-bus address 0 input ^[2]
	MOSI	I/O	SPI master out, slave in ^[2]
	MX	O	UART output to microcontroller ^[2]
31	D7	I/O	test port
	SCL	I/O	I ² C-bus clock input/output ^[2]
	MISO	I/O	SPI master in, slave out ^[2]
	TX	O	UART data output to microcontroller ^[2]
32	EA	I	external address input for coding I ² C-bus address ^[2]

[1] Pin types: I = Input, O = Output, I/O = Input/Output, P = Power and G = Ground.

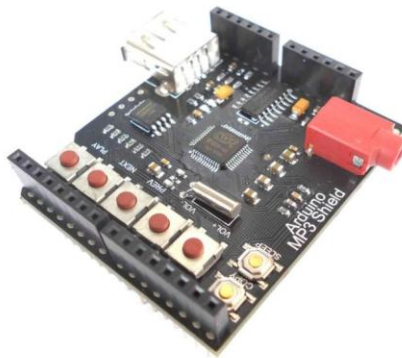
[2] The pin functionality of these pins is explained in [Section 8.1 "Digital interfaces"](#).

[3] Connection of heatsink pad on package bottom side is not necessary. Optional connection to pin DVSS is possible.

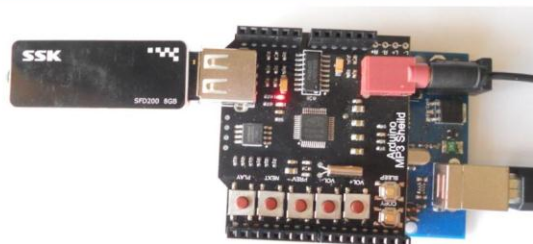
3. MP3 Shield

www.elechouse.com

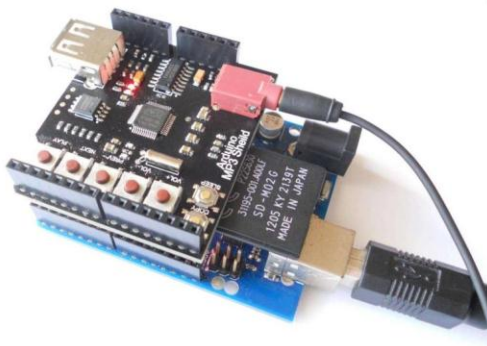
Arduino MP3 Shield User Guide



Instruction



Arduino MP3 Shield is a high quality MP3 module, supporting standard MP3 and WAV format audio files. It also supports three kinds of storage devices, including an onboard SPI-Flash, external SD Card and U-Disk.



The Arduino MP3 Shield has an onboard 3W Class-D stereo audio amplifier which could drive 4ohm load such as loudspeaker.

The Arduino MP3 Shield has seven buttons and a UART interface, that is to say user has two ways to control the module flexibly. The UART interface is connected to the Arduino hardware UART by default. Since most Arduino boards have only one UART interface, this module's UART interface can be redefined to Arduino D7 (TX) and D8 (RX) through the jumpers on board. With SoftSerial lib, Arduino could also send data to this module.

We tested with 8G SDHC card and 16G U-disk. It works in perfect compatibility. We haven't tested larger SD and U-disk yet.

Feature

- Plug in and play for Arduino
- Control by Serial UART interface or keys onboard.
- Directly drive 32ohm headset.
- On-board Audio Amplifier to drive 3W/4Ω external loudspeaker.
- 32 levels adjustable volume.
- Support FAT16/FAT32 file system.
- Support up to SD/SDHC card and U-Disk.
- On board 64Mbit SPI-FLASH.
- Support copying audio files from SD card or U-Disk to SPI-FLASH.
- Supporting Sleep Mode.
- Support 16K~320Kbps MP3 files and 8K~44.1KHz sampling rate WAV files (SPI-FLASH only supports MP3 format)

Parameter

- Power Supply: 4.5V~5.5V, typically 5V.
- Interface: UART (5V TTL) and 7 buttons.
- Size: 47mm x 53mm.

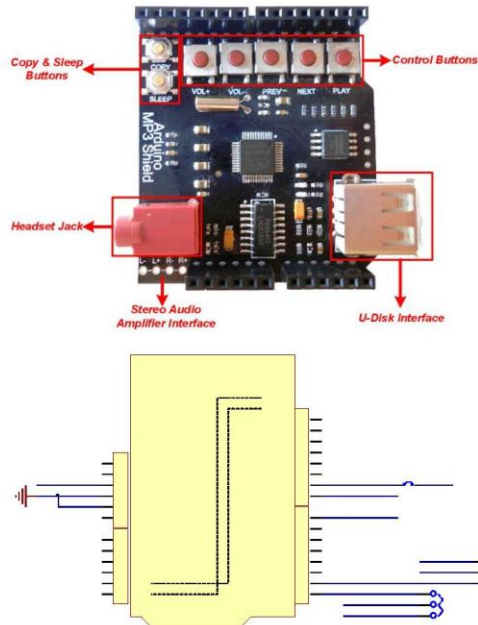
LED

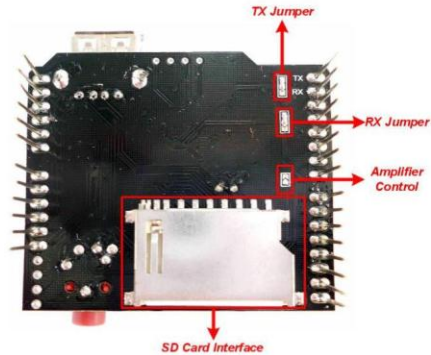
There are 2 LEDs on the board.

- Power indicating: the LED is always on when power is supplied.
- State indicating: LED is on when playing music, blinking when copying files to onboard SPI-FLASH.

Function Description

On board Audio Amplifier could drive up to 3W/4 Ω external loudspeaker. Infact, we test it with 5W/8 Ω loudspeaker, and it works well.





By default, the RX Jumper and TX Jumper are connected to hardware UART pins on Arduino. You could change it with soldering tool. If you bridge the lower two pads, Arduino D7 and D8 need to serve as RX and TX. You could find more information in latter description.

The amplifier could be controlled by Arduino. By default, it is enabled and not controlled by Arduino. If you need, please bright the Amplifier Control pads. More information will be explained later.

CONFIGURATION DESCRIPTION

Arduino MP3 Shield configuration is stored in a file named "iSound.mp3" (Note that in fact it is not an audio file but a txt file. You could open it with software such as Notepad). The format of the "iSound.mp3" is as bellows:

```
spxx//
clxx//
cfxx xx xx xx//
```

Details:

spxx// Play settings when powered on. *sp* must be lower case letters, *xx* must be 00 or 01. **00** means "Do not play automatically when powered on". **01** means "Play automatically when power on".

clxx// Play Mode Settings. *cl* must be lower case letters, *xx* must be 00, 01, 02 or 03. **00** means **Play single music**, **01** means **Repeat track**, **02** means **Repeat all**, and **03** means **Play randomly**.

cfxx xx xx xx// User-defined data. User can read these four bytes data through UART interface. Refer to [Get user-defined data](#).

"iSound.mp3" is a file where you could configure the Arduino MP3 Shield. Follow the steps below to configure it:

Create a text file named "iSound.mp3" on your PC. Usually open the Notepad in Windows (on Mac or Linux, see the note), create a new file, enter the data in the rule above, and save as mp3 file.

KEY**Standard Key**

KEY	OPERATION	FUNCTION
PLAY	SHORT	Play/Pause music
	LONG	Stop playing
PREV	SHORT	Previous music
	LONG	Fast forward (when playing music)
NEXT	SHORT	Next music
	LONG	Fast backward(when plying music)
VOL-	SHORT	Volume -
	LONG	Volume- fast
VOL+	SHORT	Volume +
	LONG	Volume + fast

COPY and SLEEP

The two bronze head button on the upper left corner are COPY and SLEEP button.

COPY Button

The basis function of COPY button is copy music files to SPI-FLASH. For more information, please refer to [Copy Music through COPY button](#).

Another function of this COPY button is to play next. It functions while this module is the stage of playing music.

SLEEP

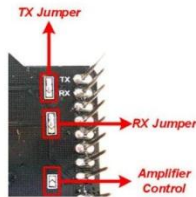
Arduino MP3 Shield can sleep or wake up by SLEEP button.

SLEEP: Press SLEEP button and hold 2 more seconds.

WAKE UP: Press SLEEP button to generate a high level pulse to wakeup Arduino MP3 Shield.

JUMPERS

On the bottom of the Arduino MP3 Shield, there are three jumpers, **TX Jumper**, **RX Jumper**, and **Amplifier Control Jumper**. They are used to select alternate function of Arduino MP3 Shield.

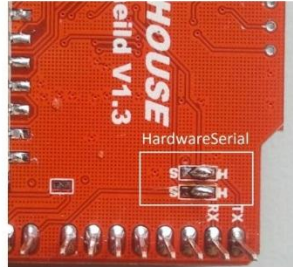
**TX and RX jumper**

By default, the Arduino MP3 Shield connects Arduino board with **hardware UART** through the jumper. It is very easy to program if the hardware UART is used. However, this might cause problems while programming the Arduino with this MP3 shield on it. Since programming will use the hardware UART interface while it is occupied by the MP3 shield, you may get errors and fail to upload code. So in some cases, Softserial control is useful. To change the connection, dis-bridge upper two pads and bridge the two below. Then on this module, D7 becomes TX, D8 becomes RX. **While programming with SoftSerial library on Arduino, Arduino D7 should serve as RX and Arduino D8 as TX.**

Now SoftSerial-controlling code is in our library. You could find the example.



Software Serial



Hardware Serial

Amplifier Control

Amplifier Control pin is used to enable or disable the on board amplifier. By default, the amplifier is enabled, and the control pin is disconnected. To control the amplifier, bridge the two pads of the Amplifier Control Jumper in the picture above. Then the control pin is connected to **Arduino D9**. Set D9 **high** to **enable** the on board amplifier, **low** to **shutdown** the amplifier.

Command Description

Through UART interface, user can send command to select music, select storage, set play mode, inter-cut music, music combination play and so on.

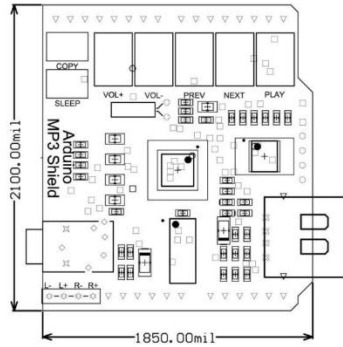
UART Configuration

UART setting:

- Baud Rate: 9600bps
- Parity bit: None
- Start bit: 1
- Data bit: 8
- Stop bit: 1

Command Format

START	LENGTH	COMMAND	PARAMETER	END
0x7E				0x7E



Code

Download the library: [MP3_shield_library.zip](#)

Disclaimer and Revisions

The information in this document may change without notice. Please visit [www.elechouse.com](#) for new information.

Revision History

Rev.	Date	Author	Description
A	July 3 rd , 2012	Lich	Initial version
B	August 23, 2012	Wilson	Add the mapping pin
C	Oct. 28 th , 2012	Wilson	Add note for the iSound file
D	Dec. 12 th , 2012	Wilson	Add library supporting Softserial control

-----Halaman ini sengaja dikosongkan-----

D. Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



Nama : Cahyo Dhian Tyastono
TTL : Trenggalek, 9 Desember 1995
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. K. Pattimura Gg. Orak Arik no. 2 Ngantru, Trenggalek
Telp/HP : 085105757272
E-mail : cahyodhian.cdt@gmail.com

RIWAYAT PENDIDIKAN:

1. 2002-2008 : SD Negeri 3 Ngantru Trenggalek
2. 2008-2011 : SMP Negeri 1 Trenggalek
3. 2011-2014 : SMA Negeri 1 Trenggalek
4. 2014-2017 : D3 Teknik Elektro, Program Studi Komputer Kontrol – Fakultas Vokasi

PENGALAMAN KERJA

1. Kerja Praktek di PT. Terminal Teluk Lamong Surabaya

PENGALAMAN ORGANISASI

1. Staff Kewirausahaan Badan Eksekutif Mahasiswa FTI-ITS 2015-2016
2. Kabiرو Fund Raising Kewirausahaan Himpunan Mahasiswa D3 Teknik Elektro ITS 2016-2017

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Dany Prasetyo
TTL : Banyuwangi, 11 April 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Griya Masangan Wetan No. B1-02
Telp/HP : 08973696055
E-mail : dhaniolx@gmail.com

RIWAYAT PENDIDIKAN:

1. 2002-2008 : SD Negeri Wage 1 Taman
2. 2008-2011 : SMP YPM 1 Taman
3. 2011-2014 : SMK YPM 1 Taman
4. 2014-2017 : D3 Teknik Elektro, Program Studi Komputer Kontrol – Fakultas Vokasi

PENGALAMAN KERJA

1. Kerja Praktek di PT. Pokphand Indonesia Surabaya

PENGALAMAN KEPANITIAN

1. Staff Smart Robotic Car IARC 2015/2016
2. OC Gerigi ITS 2015/2016
3. OC LKMM PRA-TD XII FTI-ITS 2015/2016

-----Halaman ini sengaja dikosongkan-----